Вариант: 1-1-1

1. gcd(-88, 68) = 4

1 def gcd(x=-88, y=68)

2 if -88 < 0: --- True

3 x = --88

x = 88

4 if 68 < 0: --- False

6 while 68 != 0: --- True

7 rem = 88 % 68

rem = 20

8 x = 68

9 y = 20

6 while 20 != 0: --- True

7 rem = 68 % 20

rem = 8

8 x = 20

9 y = 8

6 while 8 != 0: --- True

7 rem = 20 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(78, 0) = 78

1 def gcd(x=78, y=0)

2 if 78 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 78

3. hex(230) = 'E6'

1 def hex(number=230)

2 if 230 == 0: --- False

4 res = ''

5 while 230 > 0: --- True

6 digit = 230 % 16

digit = 6

7 if 6 <= 9: --- True

8 digit = str(6)

digit = '6'

23 res = '6' + ''

res = '6'

24 number = 230 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '6'

res = 'E6'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E6'

4. square\_equal(-3, 0, 48) = '4.0 and -4.0'

3 def square\_equal(a=-3, b=0, c=48)

4 if -3 != 0: --- True

5 D = 0\*0 - 4\*-3\*48

D = 576

6 if 576 > 0: --- True

7 x1 = (-0 - sqrt(576)) / (2\*-3)

x1 = 4.0

8 x2 = (-0 + sqrt(576)) / (2\*-3)

x2 = -4.0

9 return str(4.0) + ' and ' + str(-4.0)

return '4.0 and -4.0'

5. square\_equal(-86, -42, -7) = 'no roots'

3 def square\_equal(a=-86, b=-42, c=-7)

4 if -86 != 0: --- True

5 D = -42\*-42 - 4\*-86\*-7

D = -644

6 if -644 > 0: --- False

10 elif -644 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(2) = '2'

1 def factorize(n=2)

2 res = ''

3 while 2 > 2 and 2 % 2 == 0: --- False

6 d = 3

7 while 2 > 3: --- False

13 return '' + str(2)

return '2'

7. remove\_digit(543, 4) = 53

1 def remove\_digit(number=543, digit=4)

2 res = 0

3 power = 1

4 while 543 > 0: --- True

5 cur\_digit = 543 % 10

cur\_digit = 3

6 if 3 != 4: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 543 // 10

number = 54

4 while 54 > 0: --- True

5 cur\_digit = 54 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 54 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 4: --- True

7 res = 3 + 5 \* 10

res = 53

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 53

Вариант: 1-1-2

1. gcd(-39, -96) = 3

1 def gcd(x=-39, y=-96)

2 if -39 < 0: --- True

3 x = --39

x = 39

4 if -96 < 0: --- True

5 y = --96

y = 96

6 while 96 != 0: --- True

7 rem = 39 % 96

rem = 39

8 x = 96

9 y = 39

6 while 39 != 0: --- True

7 rem = 96 % 39

rem = 18

8 x = 39

9 y = 18

6 while 18 != 0: --- True

7 rem = 39 % 18

rem = 3

8 x = 18

9 y = 3

6 while 3 != 0: --- True

7 rem = 18 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(-88, 0) = 88

1 def gcd(x=-88, y=0)

2 if -88 < 0: --- True

3 x = --88

x = 88

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 88

3. hex(175) = 'AF'

1 def hex(number=175)

2 if 175 == 0: --- False

4 res = ''

5 while 175 > 0: --- True

6 digit = 175 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + ''

res = 'F'

24 number = 175 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + 'F'

res = 'AF'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'AF'

4. square\_equal(4, 85, -89) = '-22.25 and 1.0'

3 def square\_equal(a=4, b=85, c=-89)

4 if 4 != 0: --- True

5 D = 85\*85 - 4\*4\*-89

D = 8649

6 if 8649 > 0: --- True

7 x1 = (-85 - sqrt(8649)) / (2\*4)

x1 = -22.25

8 x2 = (-85 + sqrt(8649)) / (2\*4)

x2 = 1.0

9 return str(-22.25) + ' and ' + str(1.0)

return '-22.25 and 1.0'

5. square\_equal(-14, 29, -100) = 'no roots'

3 def square\_equal(a=-14, b=29, c=-100)

4 if -14 != 0: --- True

5 D = 29\*29 - 4\*-14\*-100

D = -4759

6 if -4759 > 0: --- False

10 elif -4759 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(250) = '2\*5\*5\*5'

1 def factorize(n=250)

2 res = ''

3 while 250 > 2 and 250 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 250 // 2

n = 125

3 while 125 > 2 and 125 % 2 == 0: --- False

6 d = 3

7 while 125 > 3: --- True

8 if 125 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 125 > 5: --- True

8 if 125 % 5 == 0: --- True

9 res = '2\*' + str(5) + '\*'

res = '2\*5\*'

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '2\*5\*' + str(5) + '\*'

res = '2\*5\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '2\*5\*5\*' + str(5)

return '2\*5\*5\*5'

7. remove\_digit(406, 6) = 40

1 def remove\_digit(number=406, digit=6)

2 res = 0

3 power = 1

4 while 406 > 0: --- True

5 cur\_digit = 406 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 406 // 10

number = 40

4 while 40 > 0: --- True

5 cur\_digit = 40 % 10

cur\_digit = 0

6 if 0 != 6: --- True

7 res = 0 + 0 \* 1

res = 0

8 power = 1 \* 10

power = 10

9 number = 40 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 6: --- True

7 res = 0 + 4 \* 10

res = 40

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 40

Вариант: 1-1-3

1. gcd(-39, -33) = 3

1 def gcd(x=-39, y=-33)

2 if -39 < 0: --- True

3 x = --39

x = 39

4 if -33 < 0: --- True

5 y = --33

y = 33

6 while 33 != 0: --- True

7 rem = 39 % 33

rem = 6

8 x = 33

9 y = 6

6 while 6 != 0: --- True

7 rem = 33 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(-23, 0) = 23

1 def gcd(x=-23, y=0)

2 if -23 < 0: --- True

3 x = --23

x = 23

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 23

3. hex(199) = 'C7'

1 def hex(number=199)

2 if 199 == 0: --- False

4 res = ''

5 while 199 > 0: --- True

6 digit = 199 % 16

digit = 7

7 if 7 <= 9: --- True

8 digit = str(7)

digit = '7'

23 res = '7' + ''

res = '7'

24 number = 199 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '7'

res = 'C7'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C7'

4. square\_equal(23, -23, -46) = '-1.0 and 2.0'

3 def square\_equal(a=23, b=-23, c=-46)

4 if 23 != 0: --- True

5 D = -23\*-23 - 4\*23\*-46

D = 4761

6 if 4761 > 0: --- True

7 x1 = (--23 - sqrt(4761)) / (2\*23)

x1 = -1.0

8 x2 = (--23 + sqrt(4761)) / (2\*23)

x2 = 2.0

9 return str(-1.0) + ' and ' + str(2.0)

return '-1.0 and 2.0'

5. square\_equal(-56, 89, -54) = 'no roots'

3 def square\_equal(a=-56, b=89, c=-54)

4 if -56 != 0: --- True

5 D = 89\*89 - 4\*-56\*-54

D = -4175

6 if -4175 > 0: --- False

10 elif -4175 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(32) = '2\*2\*2\*2\*2'

1 def factorize(n=32)

2 res = ''

3 while 32 > 2 and 32 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 32 // 2

n = 16

3 while 16 > 2 and 16 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 16 // 2

n = 8

3 while 8 > 2 and 8 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 8 // 2

n = 4

3 while 4 > 2 and 4 % 2 == 0: --- True

4 res = '2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*'

5 n = 4 // 2

n = 2

3 while 2 > 2 and 2 % 2 == 0: --- False

6 d = 3

7 while 2 > 3: --- False

13 return '2\*2\*2\*2\*' + str(2)

return '2\*2\*2\*2\*2'

7. remove\_digit(829, 2) = 89

1 def remove\_digit(number=829, digit=2)

2 res = 0

3 power = 1

4 while 829 > 0: --- True

5 cur\_digit = 829 % 10

cur\_digit = 9

6 if 9 != 2: --- True

7 res = 0 + 9 \* 1

res = 9

8 power = 1 \* 10

power = 10

9 number = 829 // 10

number = 82

4 while 82 > 0: --- True

5 cur\_digit = 82 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 82 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 2: --- True

7 res = 9 + 8 \* 10

res = 89

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 89

Вариант: 1-1-4

1. gcd(-92, 48) = 4

1 def gcd(x=-92, y=48)

2 if -92 < 0: --- True

3 x = --92

x = 92

4 if 48 < 0: --- False

6 while 48 != 0: --- True

7 rem = 92 % 48

rem = 44

8 x = 48

9 y = 44

6 while 44 != 0: --- True

7 rem = 48 % 44

rem = 4

8 x = 44

9 y = 4

6 while 4 != 0: --- True

7 rem = 44 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(-54, 0) = 54

1 def gcd(x=-54, y=0)

2 if -54 < 0: --- True

3 x = --54

x = 54

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 54

3. hex(226) = 'E2'

1 def hex(number=226)

2 if 226 == 0: --- False

4 res = ''

5 while 226 > 0: --- True

6 digit = 226 % 16

digit = 2

7 if 2 <= 9: --- True

8 digit = str(2)

digit = '2'

23 res = '2' + ''

res = '2'

24 number = 226 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '2'

res = 'E2'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E2'

4. square\_equal(-2, 32, -30) = '15.0 and 1.0'

3 def square\_equal(a=-2, b=32, c=-30)

4 if -2 != 0: --- True

5 D = 32\*32 - 4\*-2\*-30

D = 784

6 if 784 > 0: --- True

7 x1 = (-32 - sqrt(784)) / (2\*-2)

x1 = 15.0

8 x2 = (-32 + sqrt(784)) / (2\*-2)

x2 = 1.0

9 return str(15.0) + ' and ' + str(1.0)

return '15.0 and 1.0'

5. square\_equal(27, -37, 43) = 'no roots'

3 def square\_equal(a=27, b=-37, c=43)

4 if 27 != 0: --- True

5 D = -37\*-37 - 4\*27\*43

D = -3275

6 if -3275 > 0: --- False

10 elif -3275 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(625) = '5\*5\*5\*5'

1 def factorize(n=625)

2 res = ''

3 while 625 > 2 and 625 % 2 == 0: --- False

6 d = 3

7 while 625 > 3: --- True

8 if 625 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 625 > 5: --- True

8 if 625 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 625 // 5

n = 125

7 while 125 > 5: --- True

8 if 125 % 5 == 0: --- True

9 res = '5\*' + str(5) + '\*'

res = '5\*5\*'

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '5\*5\*' + str(5) + '\*'

res = '5\*5\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '5\*5\*5\*' + str(5)

return '5\*5\*5\*5'

7. remove\_digit(831, 3) = 81

1 def remove\_digit(number=831, digit=3)

2 res = 0

3 power = 1

4 while 831 > 0: --- True

5 cur\_digit = 831 % 10

cur\_digit = 1

6 if 1 != 3: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 831 // 10

number = 83

4 while 83 > 0: --- True

5 cur\_digit = 83 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 83 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 3: --- True

7 res = 1 + 8 \* 10

res = 81

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 81

Вариант: 1-1-5

1. gcd(-96, 100) = 4

1 def gcd(x=-96, y=100)

2 if -96 < 0: --- True

3 x = --96

x = 96

4 if 100 < 0: --- False

6 while 100 != 0: --- True

7 rem = 96 % 100

rem = 96

8 x = 100

9 y = 96

6 while 96 != 0: --- True

7 rem = 100 % 96

rem = 4

8 x = 96

9 y = 4

6 while 4 != 0: --- True

7 rem = 96 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(11, 0) = 11

1 def gcd(x=11, y=0)

2 if 11 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 11

3. hex(215) = 'D7'

1 def hex(number=215)

2 if 215 == 0: --- False

4 res = ''

5 while 215 > 0: --- True

6 digit = 215 % 16

digit = 7

7 if 7 <= 9: --- True

8 digit = str(7)

digit = '7'

23 res = '7' + ''

res = '7'

24 number = 215 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '7'

res = 'D7'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D7'

4. square\_equal(-1, 16, -48) = '12.0 and 4.0'

3 def square\_equal(a=-1, b=16, c=-48)

4 if -1 != 0: --- True

5 D = 16\*16 - 4\*-1\*-48

D = 64

6 if 64 > 0: --- True

7 x1 = (-16 - sqrt(64)) / (2\*-1)

x1 = 12.0

8 x2 = (-16 + sqrt(64)) / (2\*-1)

x2 = 4.0

9 return str(12.0) + ' and ' + str(4.0)

return '12.0 and 4.0'

5. square\_equal(52, 55, 16) = 'no roots'

3 def square\_equal(a=52, b=55, c=16)

4 if 52 != 0: --- True

5 D = 55\*55 - 4\*52\*16

D = -303

6 if -303 > 0: --- False

10 elif -303 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(210) = '2\*3\*5\*7'

1 def factorize(n=210)

2 res = ''

3 while 210 > 2 and 210 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 210 // 2

n = 105

3 while 105 > 2 and 105 % 2 == 0: --- False

6 d = 3

7 while 105 > 3: --- True

8 if 105 % 3 == 0: --- True

9 res = '2\*' + str(3) + '\*'

res = '2\*3\*'

10 n = 105 // 3

n = 35

7 while 35 > 3: --- True

8 if 35 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 35 > 5: --- True

8 if 35 % 5 == 0: --- True

9 res = '2\*3\*' + str(5) + '\*'

res = '2\*3\*5\*'

10 n = 35 // 5

n = 7

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '2\*3\*5\*' + str(7)

return '2\*3\*5\*7'

7. remove\_digit(915, 5) = 91

1 def remove\_digit(number=915, digit=5)

2 res = 0

3 power = 1

4 while 915 > 0: --- True

5 cur\_digit = 915 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 915 // 10

number = 91

4 while 91 > 0: --- True

5 cur\_digit = 91 % 10

cur\_digit = 1

6 if 1 != 5: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 91 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 5: --- True

7 res = 1 + 9 \* 10

res = 91

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 91

Вариант: 1-1-6

1. gcd(18, 27) = 9

1 def gcd(x=18, y=27)

2 if 18 < 0: --- False

4 if 27 < 0: --- False

6 while 27 != 0: --- True

7 rem = 18 % 27

rem = 18

8 x = 27

9 y = 18

6 while 18 != 0: --- True

7 rem = 27 % 18

rem = 9

8 x = 18

9 y = 9

6 while 9 != 0: --- True

7 rem = 18 % 9

rem = 0

8 x = 9

9 y = 0

6 while 0 != 0: --- False

10 return 9

2. gcd(22, 0) = 22

1 def gcd(x=22, y=0)

2 if 22 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 22

3. hex(214) = 'D6'

1 def hex(number=214)

2 if 214 == 0: --- False

4 res = ''

5 while 214 > 0: --- True

6 digit = 214 % 16

digit = 6

7 if 6 <= 9: --- True

8 digit = str(6)

digit = '6'

23 res = '6' + ''

res = '6'

24 number = 214 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '6'

res = 'D6'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D6'

4. square\_equal(-20, 44, 27) = '2.7 and -0.5'

3 def square\_equal(a=-20, b=44, c=27)

4 if -20 != 0: --- True

5 D = 44\*44 - 4\*-20\*27

D = 4096

6 if 4096 > 0: --- True

7 x1 = (-44 - sqrt(4096)) / (2\*-20)

x1 = 2.7

8 x2 = (-44 + sqrt(4096)) / (2\*-20)

x2 = -0.5

9 return str(2.7) + ' and ' + str(-0.5)

return '2.7 and -0.5'

5. square\_equal(-17, 15, -26) = 'no roots'

3 def square\_equal(a=-17, b=15, c=-26)

4 if -17 != 0: --- True

5 D = 15\*15 - 4\*-17\*-26

D = -1543

6 if -1543 > 0: --- False

10 elif -1543 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(50) = '2\*5\*5'

1 def factorize(n=50)

2 res = ''

3 while 50 > 2 and 50 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 50 // 2

n = 25

3 while 25 > 2 and 25 % 2 == 0: --- False

6 d = 3

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '2\*' + str(5) + '\*'

res = '2\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '2\*5\*' + str(5)

return '2\*5\*5'

7. remove\_digit(790, 9) = 70

1 def remove\_digit(number=790, digit=9)

2 res = 0

3 power = 1

4 while 790 > 0: --- True

5 cur\_digit = 790 % 10

cur\_digit = 0

6 if 0 != 9: --- True

7 res = 0 + 0 \* 1

res = 0

8 power = 1 \* 10

power = 10

9 number = 790 // 10

number = 79

4 while 79 > 0: --- True

5 cur\_digit = 79 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 79 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 9: --- True

7 res = 0 + 7 \* 10

res = 70

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 70

Вариант: 1-1-7

1. gcd(-66, 96) = 6

1 def gcd(x=-66, y=96)

2 if -66 < 0: --- True

3 x = --66

x = 66

4 if 96 < 0: --- False

6 while 96 != 0: --- True

7 rem = 66 % 96

rem = 66

8 x = 96

9 y = 66

6 while 66 != 0: --- True

7 rem = 96 % 66

rem = 30

8 x = 66

9 y = 30

6 while 30 != 0: --- True

7 rem = 66 % 30

rem = 6

8 x = 30

9 y = 6

6 while 6 != 0: --- True

7 rem = 30 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(95, 0) = 95

1 def gcd(x=95, y=0)

2 if 95 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 95

3. hex(186) = 'BA'

1 def hex(number=186)

2 if 186 == 0: --- False

4 res = ''

5 while 186 > 0: --- True

6 digit = 186 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + ''

res = 'A'

24 number = 186 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + 'A'

res = 'BA'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'BA'

4. square\_equal(36, 45, 9) = '-1.0 and -0.25'

3 def square\_equal(a=36, b=45, c=9)

4 if 36 != 0: --- True

5 D = 45\*45 - 4\*36\*9

D = 729

6 if 729 > 0: --- True

7 x1 = (-45 - sqrt(729)) / (2\*36)

x1 = -1.0

8 x2 = (-45 + sqrt(729)) / (2\*36)

x2 = -0.25

9 return str(-1.0) + ' and ' + str(-0.25)

return '-1.0 and -0.25'

5. square\_equal(21, 28, 19) = 'no roots'

3 def square\_equal(a=21, b=28, c=19)

4 if 21 != 0: --- True

5 D = 28\*28 - 4\*21\*19

D = -812

6 if -812 > 0: --- False

10 elif -812 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(120) = '2\*2\*2\*3\*5'

1 def factorize(n=120)

2 res = ''

3 while 120 > 2 and 120 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 120 // 2

n = 60

3 while 60 > 2 and 60 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 60 // 2

n = 30

3 while 30 > 2 and 30 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 30 // 2

n = 15

3 while 15 > 2 and 15 % 2 == 0: --- False

6 d = 3

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '2\*2\*2\*' + str(3) + '\*'

res = '2\*2\*2\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*2\*3\*' + str(5)

return '2\*2\*2\*3\*5'

7. remove\_digit(504, 0) = 54

1 def remove\_digit(number=504, digit=0)

2 res = 0

3 power = 1

4 while 504 > 0: --- True

5 cur\_digit = 504 % 10

cur\_digit = 4

6 if 4 != 0: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 504 // 10

number = 50

4 while 50 > 0: --- True

5 cur\_digit = 50 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 50 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 0: --- True

7 res = 4 + 5 \* 10

res = 54

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 54

Вариант: 1-1-8

1. gcd(-8, -60) = 4

1 def gcd(x=-8, y=-60)

2 if -8 < 0: --- True

3 x = --8

x = 8

4 if -60 < 0: --- True

5 y = --60

y = 60

6 while 60 != 0: --- True

7 rem = 8 % 60

rem = 8

8 x = 60

9 y = 8

6 while 8 != 0: --- True

7 rem = 60 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(-64, 0) = 64

1 def gcd(x=-64, y=0)

2 if -64 < 0: --- True

3 x = --64

x = 64

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 64

3. hex(171) = 'AB'

1 def hex(number=171)

2 if 171 == 0: --- False

4 res = ''

5 while 171 > 0: --- True

6 digit = 171 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + ''

res = 'B'

24 number = 171 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + 'B'

res = 'AB'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'AB'

4. square\_equal(36, 18, -54) = '-1.5 and 1.0'

3 def square\_equal(a=36, b=18, c=-54)

4 if 36 != 0: --- True

5 D = 18\*18 - 4\*36\*-54

D = 8100

6 if 8100 > 0: --- True

7 x1 = (-18 - sqrt(8100)) / (2\*36)

x1 = -1.5

8 x2 = (-18 + sqrt(8100)) / (2\*36)

x2 = 1.0

9 return str(-1.5) + ' and ' + str(1.0)

return '-1.5 and 1.0'

5. square\_equal(-58, 100, -57) = 'no roots'

3 def square\_equal(a=-58, b=100, c=-57)

4 if -58 != 0: --- True

5 D = 100\*100 - 4\*-58\*-57

D = -3224

6 if -3224 > 0: --- False

10 elif -3224 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(294) = '2\*3\*7\*7'

1 def factorize(n=294)

2 res = ''

3 while 294 > 2 and 294 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 294 // 2

n = 147

3 while 147 > 2 and 147 % 2 == 0: --- False

6 d = 3

7 while 147 > 3: --- True

8 if 147 % 3 == 0: --- True

9 res = '2\*' + str(3) + '\*'

res = '2\*3\*'

10 n = 147 // 3

n = 49

7 while 49 > 3: --- True

8 if 49 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 49 > 5: --- True

8 if 49 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '2\*3\*' + str(7) + '\*'

res = '2\*3\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '2\*3\*7\*' + str(7)

return '2\*3\*7\*7'

7. remove\_digit(126, 2) = 16

1 def remove\_digit(number=126, digit=2)

2 res = 0

3 power = 1

4 while 126 > 0: --- True

5 cur\_digit = 126 % 10

cur\_digit = 6

6 if 6 != 2: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 126 // 10

number = 12

4 while 12 > 0: --- True

5 cur\_digit = 12 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 12 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 2: --- True

7 res = 6 + 1 \* 10

res = 16

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 16

Вариант: 1-1-9

1. gcd(-30, 42) = 6

1 def gcd(x=-30, y=42)

2 if -30 < 0: --- True

3 x = --30

x = 30

4 if 42 < 0: --- False

6 while 42 != 0: --- True

7 rem = 30 % 42

rem = 30

8 x = 42

9 y = 30

6 while 30 != 0: --- True

7 rem = 42 % 30

rem = 12

8 x = 30

9 y = 12

6 while 12 != 0: --- True

7 rem = 30 % 12

rem = 6

8 x = 12

9 y = 6

6 while 6 != 0: --- True

7 rem = 12 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(0, 85) = 85

1 def gcd(x=0, y=85)

2 if 0 < 0: --- False

4 if 85 < 0: --- False

6 while 85 != 0: --- True

7 rem = 0 % 85

rem = 0

8 x = 85

9 y = 0

6 while 0 != 0: --- False

10 return 85

3. hex(222) = 'DE'

1 def hex(number=222)

2 if 222 == 0: --- False

4 res = ''

5 while 222 > 0: --- True

6 digit = 222 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + ''

res = 'E'

24 number = 222 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + 'E'

res = 'DE'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DE'

4. square\_equal(16, -88, -23) = '-0.25 and 5.75'

3 def square\_equal(a=16, b=-88, c=-23)

4 if 16 != 0: --- True

5 D = -88\*-88 - 4\*16\*-23

D = 9216

6 if 9216 > 0: --- True

7 x1 = (--88 - sqrt(9216)) / (2\*16)

x1 = -0.25

8 x2 = (--88 + sqrt(9216)) / (2\*16)

x2 = 5.75

9 return str(-0.25) + ' and ' + str(5.75)

return '-0.25 and 5.75'

5. square\_equal(10, -54, 76) = 'no roots'

3 def square\_equal(a=10, b=-54, c=76)

4 if 10 != 0: --- True

5 D = -54\*-54 - 4\*10\*76

D = -124

6 if -124 > 0: --- False

10 elif -124 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(60) = '2\*2\*3\*5'

1 def factorize(n=60)

2 res = ''

3 while 60 > 2 and 60 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 60 // 2

n = 30

3 while 30 > 2 and 30 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 30 // 2

n = 15

3 while 15 > 2 and 15 % 2 == 0: --- False

6 d = 3

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '2\*2\*' + str(3) + '\*'

res = '2\*2\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*3\*' + str(5)

return '2\*2\*3\*5'

7. remove\_digit(651, 5) = 61

1 def remove\_digit(number=651, digit=5)

2 res = 0

3 power = 1

4 while 651 > 0: --- True

5 cur\_digit = 651 % 10

cur\_digit = 1

6 if 1 != 5: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 651 // 10

number = 65

4 while 65 > 0: --- True

5 cur\_digit = 65 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 65 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 5: --- True

7 res = 1 + 6 \* 10

res = 61

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 61

Вариант: 1-1-10

1. gcd(48, -63) = 3

1 def gcd(x=48, y=-63)

2 if 48 < 0: --- False

4 if -63 < 0: --- True

5 y = --63

y = 63

6 while 63 != 0: --- True

7 rem = 48 % 63

rem = 48

8 x = 63

9 y = 48

6 while 48 != 0: --- True

7 rem = 63 % 48

rem = 15

8 x = 48

9 y = 15

6 while 15 != 0: --- True

7 rem = 48 % 15

rem = 3

8 x = 15

9 y = 3

6 while 3 != 0: --- True

7 rem = 15 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(71, 0) = 71

1 def gcd(x=71, y=0)

2 if 71 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 71

3. hex(162) = 'A2'

1 def hex(number=162)

2 if 162 == 0: --- False

4 res = ''

5 while 162 > 0: --- True

6 digit = 162 % 16

digit = 2

7 if 2 <= 9: --- True

8 digit = str(2)

digit = '2'

23 res = '2' + ''

res = '2'

24 number = 162 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '2'

res = 'A2'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A2'

4. square\_equal(0, 58, -87) = '1.5'

3 def square\_equal(a=0, b=58, c=-87)

4 if 0 != 0: --- False

14 else:

15 if 58 != 0: --- True

16 return str(--87 / 58)

return '1.5'

5. square\_equal(73, 2, 1) = 'no roots'

3 def square\_equal(a=73, b=2, c=1)

4 if 73 != 0: --- True

5 D = 2\*2 - 4\*73\*1

D = -288

6 if -288 > 0: --- False

10 elif -288 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(500) = '2\*2\*5\*5\*5'

1 def factorize(n=500)

2 res = ''

3 while 500 > 2 and 500 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 500 // 2

n = 250

3 while 250 > 2 and 250 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 250 // 2

n = 125

3 while 125 > 2 and 125 % 2 == 0: --- False

6 d = 3

7 while 125 > 3: --- True

8 if 125 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 125 > 5: --- True

8 if 125 % 5 == 0: --- True

9 res = '2\*2\*' + str(5) + '\*'

res = '2\*2\*5\*'

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '2\*2\*5\*' + str(5) + '\*'

res = '2\*2\*5\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '2\*2\*5\*5\*' + str(5)

return '2\*2\*5\*5\*5'

7. remove\_digit(374, 7) = 34

1 def remove\_digit(number=374, digit=7)

2 res = 0

3 power = 1

4 while 374 > 0: --- True

5 cur\_digit = 374 % 10

cur\_digit = 4

6 if 4 != 7: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 374 // 10

number = 37

4 while 37 > 0: --- True

5 cur\_digit = 37 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 37 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 7: --- True

7 res = 4 + 3 \* 10

res = 34

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 34

Вариант: 1-1-11

1. gcd(-72, 76) = 4

1 def gcd(x=-72, y=76)

2 if -72 < 0: --- True

3 x = --72

x = 72

4 if 76 < 0: --- False

6 while 76 != 0: --- True

7 rem = 72 % 76

rem = 72

8 x = 76

9 y = 72

6 while 72 != 0: --- True

7 rem = 76 % 72

rem = 4

8 x = 72

9 y = 4

6 while 4 != 0: --- True

7 rem = 72 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 60) = 60

1 def gcd(x=0, y=60)

2 if 0 < 0: --- False

4 if 60 < 0: --- False

6 while 60 != 0: --- True

7 rem = 0 % 60

rem = 0

8 x = 60

9 y = 0

6 while 0 != 0: --- False

10 return 60

3. hex(219) = 'DB'

1 def hex(number=219)

2 if 219 == 0: --- False

4 res = ''

5 while 219 > 0: --- True

6 digit = 219 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + ''

res = 'B'

24 number = 219 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + 'B'

res = 'DB'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DB'

4. square\_equal(-8, 26, 70) = '5.0 and -1.75'

3 def square\_equal(a=-8, b=26, c=70)

4 if -8 != 0: --- True

5 D = 26\*26 - 4\*-8\*70

D = 2916

6 if 2916 > 0: --- True

7 x1 = (-26 - sqrt(2916)) / (2\*-8)

x1 = 5.0

8 x2 = (-26 + sqrt(2916)) / (2\*-8)

x2 = -1.75

9 return str(5.0) + ' and ' + str(-1.75)

return '5.0 and -1.75'

5. square\_equal(61, -78, 49) = 'no roots'

3 def square\_equal(a=61, b=-78, c=49)

4 if 61 != 0: --- True

5 D = -78\*-78 - 4\*61\*49

D = -5872

6 if -5872 > 0: --- False

10 elif -5872 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(63) = '3\*3\*7'

1 def factorize(n=63)

2 res = ''

3 while 63 > 2 and 63 % 2 == 0: --- False

6 d = 3

7 while 63 > 3: --- True

8 if 63 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 63 // 3

n = 21

7 while 21 > 3: --- True

8 if 21 % 3 == 0: --- True

9 res = '3\*' + str(3) + '\*'

res = '3\*3\*'

10 n = 21 // 3

n = 7

7 while 7 > 3: --- True

8 if 7 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '3\*3\*' + str(7)

return '3\*3\*7'

7. remove\_digit(954, 4) = 95

1 def remove\_digit(number=954, digit=4)

2 res = 0

3 power = 1

4 while 954 > 0: --- True

5 cur\_digit = 954 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 954 // 10

number = 95

4 while 95 > 0: --- True

5 cur\_digit = 95 % 10

cur\_digit = 5

6 if 5 != 4: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 95 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 4: --- True

7 res = 5 + 9 \* 10

res = 95

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 95

Вариант: 1-1-12

1. gcd(44, 99) = 11

1 def gcd(x=44, y=99)

2 if 44 < 0: --- False

4 if 99 < 0: --- False

6 while 99 != 0: --- True

7 rem = 44 % 99

rem = 44

8 x = 99

9 y = 44

6 while 44 != 0: --- True

7 rem = 99 % 44

rem = 11

8 x = 44

9 y = 11

6 while 11 != 0: --- True

7 rem = 44 % 11

rem = 0

8 x = 11

9 y = 0

6 while 0 != 0: --- False

10 return 11

2. gcd(79, 0) = 79

1 def gcd(x=79, y=0)

2 if 79 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 79

3. hex(232) = 'E8'

1 def hex(number=232)

2 if 232 == 0: --- False

4 res = ''

5 while 232 > 0: --- True

6 digit = 232 % 16

digit = 8

7 if 8 <= 9: --- True

8 digit = str(8)

digit = '8'

23 res = '8' + ''

res = '8'

24 number = 232 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '8'

res = 'E8'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E8'

4. square\_equal(25, -20, -21) = '-0.6 and 1.4'

3 def square\_equal(a=25, b=-20, c=-21)

4 if 25 != 0: --- True

5 D = -20\*-20 - 4\*25\*-21

D = 2500

6 if 2500 > 0: --- True

7 x1 = (--20 - sqrt(2500)) / (2\*25)

x1 = -0.6

8 x2 = (--20 + sqrt(2500)) / (2\*25)

x2 = 1.4

9 return str(-0.6) + ' and ' + str(1.4)

return '-0.6 and 1.4'

5. square\_equal(10, -3, 86) = 'no roots'

3 def square\_equal(a=10, b=-3, c=86)

4 if 10 != 0: --- True

5 D = -3\*-3 - 4\*10\*86

D = -3431

6 if -3431 > 0: --- False

10 elif -3431 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(0) = '0'

1 def factorize(n=0)

2 res = ''

3 while 0 > 2 and 0 % 2 == 0: --- False

6 d = 3

7 while 0 > 3: --- False

13 return '' + str(0)

return '0'

7. remove\_digit(712, 1) = 72

1 def remove\_digit(number=712, digit=1)

2 res = 0

3 power = 1

4 while 712 > 0: --- True

5 cur\_digit = 712 % 10

cur\_digit = 2

6 if 2 != 1: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 712 // 10

number = 71

4 while 71 > 0: --- True

5 cur\_digit = 71 % 10

cur\_digit = 1

6 if 1 != 1: --- False

9 number = 71 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 1: --- True

7 res = 2 + 7 \* 10

res = 72

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 72

Вариант: 1-1-13

1. gcd(32, -68) = 4

1 def gcd(x=32, y=-68)

2 if 32 < 0: --- False

4 if -68 < 0: --- True

5 y = --68

y = 68

6 while 68 != 0: --- True

7 rem = 32 % 68

rem = 32

8 x = 68

9 y = 32

6 while 32 != 0: --- True

7 rem = 68 % 32

rem = 4

8 x = 32

9 y = 4

6 while 4 != 0: --- True

7 rem = 32 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(17, 0) = 17

1 def gcd(x=17, y=0)

2 if 17 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 17

3. hex(250) = 'FA'

1 def hex(number=250)

2 if 250 == 0: --- False

4 res = ''

5 while 250 > 0: --- True

6 digit = 250 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + ''

res = 'A'

24 number = 250 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + 'A'

res = 'FA'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'FA'

4. square\_equal(20, 81, 70) = '-2.8 and -1.25'

3 def square\_equal(a=20, b=81, c=70)

4 if 20 != 0: --- True

5 D = 81\*81 - 4\*20\*70

D = 961

6 if 961 > 0: --- True

7 x1 = (-81 - sqrt(961)) / (2\*20)

x1 = -2.8

8 x2 = (-81 + sqrt(961)) / (2\*20)

x2 = -1.25

9 return str(-2.8) + ' and ' + str(-1.25)

return '-2.8 and -1.25'

5. square\_equal(25, 2, 86) = 'no roots'

3 def square\_equal(a=25, b=2, c=86)

4 if 25 != 0: --- True

5 D = 2\*2 - 4\*25\*86

D = -8596

6 if -8596 > 0: --- False

10 elif -8596 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(100) = '2\*2\*5\*5'

1 def factorize(n=100)

2 res = ''

3 while 100 > 2 and 100 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 100 // 2

n = 50

3 while 50 > 2 and 50 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 50 // 2

n = 25

3 while 25 > 2 and 25 % 2 == 0: --- False

6 d = 3

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '2\*2\*' + str(5) + '\*'

res = '2\*2\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '2\*2\*5\*' + str(5)

return '2\*2\*5\*5'

7. remove\_digit(925, 2) = 95

1 def remove\_digit(number=925, digit=2)

2 res = 0

3 power = 1

4 while 925 > 0: --- True

5 cur\_digit = 925 % 10

cur\_digit = 5

6 if 5 != 2: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 925 // 10

number = 92

4 while 92 > 0: --- True

5 cur\_digit = 92 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 92 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 2: --- True

7 res = 5 + 9 \* 10

res = 95

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 95

Вариант: 1-1-14

1. gcd(-20, -84) = 4

1 def gcd(x=-20, y=-84)

2 if -20 < 0: --- True

3 x = --20

x = 20

4 if -84 < 0: --- True

5 y = --84

y = 84

6 while 84 != 0: --- True

7 rem = 20 % 84

rem = 20

8 x = 84

9 y = 20

6 while 20 != 0: --- True

7 rem = 84 % 20

rem = 4

8 x = 20

9 y = 4

6 while 4 != 0: --- True

7 rem = 20 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -27) = 27

1 def gcd(x=0, y=-27)

2 if 0 < 0: --- False

4 if -27 < 0: --- True

5 y = --27

y = 27

6 while 27 != 0: --- True

7 rem = 0 % 27

rem = 0

8 x = 27

9 y = 0

6 while 0 != 0: --- False

10 return 27

3. hex(169) = 'A9'

1 def hex(number=169)

2 if 169 == 0: --- False

4 res = ''

5 while 169 > 0: --- True

6 digit = 169 % 16

digit = 9

7 if 9 <= 9: --- True

8 digit = str(9)

digit = '9'

23 res = '9' + ''

res = '9'

24 number = 169 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '9'

res = 'A9'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A9'

4. square\_equal(-6, -75, 81) = '1.0 and -13.5'

3 def square\_equal(a=-6, b=-75, c=81)

4 if -6 != 0: --- True

5 D = -75\*-75 - 4\*-6\*81

D = 7569

6 if 7569 > 0: --- True

7 x1 = (--75 - sqrt(7569)) / (2\*-6)

x1 = 1.0

8 x2 = (--75 + sqrt(7569)) / (2\*-6)

x2 = -13.5

9 return str(1.0) + ' and ' + str(-13.5)

return '1.0 and -13.5'

5. square\_equal(23, 74, 98) = 'no roots'

3 def square\_equal(a=23, b=74, c=98)

4 if 23 != 0: --- True

5 D = 74\*74 - 4\*23\*98

D = -3540

6 if -3540 > 0: --- False

10 elif -3540 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(5) = '5'

1 def factorize(n=5)

2 res = ''

3 while 5 > 2 and 5 % 2 == 0: --- False

6 d = 3

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '' + str(5)

return '5'

7. remove\_digit(103, 0) = 13

1 def remove\_digit(number=103, digit=0)

2 res = 0

3 power = 1

4 while 103 > 0: --- True

5 cur\_digit = 103 % 10

cur\_digit = 3

6 if 3 != 0: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 103 // 10

number = 10

4 while 10 > 0: --- True

5 cur\_digit = 10 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 10 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 0: --- True

7 res = 3 + 1 \* 10

res = 13

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 13

Вариант: 1-1-15

1. gcd(44, -32) = 4

1 def gcd(x=44, y=-32)

2 if 44 < 0: --- False

4 if -32 < 0: --- True

5 y = --32

y = 32

6 while 32 != 0: --- True

7 rem = 44 % 32

rem = 12

8 x = 32

9 y = 12

6 while 12 != 0: --- True

7 rem = 32 % 12

rem = 8

8 x = 12

9 y = 8

6 while 8 != 0: --- True

7 rem = 12 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 65) = 65

1 def gcd(x=0, y=65)

2 if 0 < 0: --- False

4 if 65 < 0: --- False

6 while 65 != 0: --- True

7 rem = 0 % 65

rem = 0

8 x = 65

9 y = 0

6 while 0 != 0: --- False

10 return 65

3. hex(161) = 'A1'

1 def hex(number=161)

2 if 161 == 0: --- False

4 res = ''

5 while 161 > 0: --- True

6 digit = 161 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 161 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '1'

res = 'A1'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A1'

4. square\_equal(-8, 0, 32) = '2.0 and -2.0'

3 def square\_equal(a=-8, b=0, c=32)

4 if -8 != 0: --- True

5 D = 0\*0 - 4\*-8\*32

D = 1024

6 if 1024 > 0: --- True

7 x1 = (-0 - sqrt(1024)) / (2\*-8)

x1 = 2.0

8 x2 = (-0 + sqrt(1024)) / (2\*-8)

x2 = -2.0

9 return str(2.0) + ' and ' + str(-2.0)

return '2.0 and -2.0'

5. square\_equal(-72, 62, -27) = 'no roots'

3 def square\_equal(a=-72, b=62, c=-27)

4 if -72 != 0: --- True

5 D = 62\*62 - 4\*-72\*-27

D = -3932

6 if -3932 > 0: --- False

10 elif -3932 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(40) = '2\*2\*2\*5'

1 def factorize(n=40)

2 res = ''

3 while 40 > 2 and 40 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 40 // 2

n = 20

3 while 20 > 2 and 20 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 20 // 2

n = 10

3 while 10 > 2 and 10 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 10 // 2

n = 5

3 while 5 > 2 and 5 % 2 == 0: --- False

6 d = 3

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*2\*' + str(5)

return '2\*2\*2\*5'

7. remove\_digit(589, 9) = 58

1 def remove\_digit(number=589, digit=9)

2 res = 0

3 power = 1

4 while 589 > 0: --- True

5 cur\_digit = 589 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 589 // 10

number = 58

4 while 58 > 0: --- True

5 cur\_digit = 58 % 10

cur\_digit = 8

6 if 8 != 9: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 58 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 9: --- True

7 res = 8 + 5 \* 10

res = 58

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 58

Вариант: 1-1-16

1. gcd(-20, 96) = 4

1 def gcd(x=-20, y=96)

2 if -20 < 0: --- True

3 x = --20

x = 20

4 if 96 < 0: --- False

6 while 96 != 0: --- True

7 rem = 20 % 96

rem = 20

8 x = 96

9 y = 20

6 while 20 != 0: --- True

7 rem = 96 % 20

rem = 16

8 x = 20

9 y = 16

6 while 16 != 0: --- True

7 rem = 20 % 16

rem = 4

8 x = 16

9 y = 4

6 while 4 != 0: --- True

7 rem = 16 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(5, 0) = 5

1 def gcd(x=5, y=0)

2 if 5 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 5

3. hex(197) = 'C5'

1 def hex(number=197)

2 if 197 == 0: --- False

4 res = ''

5 while 197 > 0: --- True

6 digit = 197 % 16

digit = 5

7 if 5 <= 9: --- True

8 digit = str(5)

digit = '5'

23 res = '5' + ''

res = '5'

24 number = 197 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '5'

res = 'C5'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C5'

4. square\_equal(0, 8, 56) = '-7.0'

3 def square\_equal(a=0, b=8, c=56)

4 if 0 != 0: --- False

14 else:

15 if 8 != 0: --- True

16 return str(-56 / 8)

return '-7.0'

5. square\_equal(-27, -27, -9) = 'no roots'

3 def square\_equal(a=-27, b=-27, c=-9)

4 if -27 != 0: --- True

5 D = -27\*-27 - 4\*-27\*-9

D = -243

6 if -243 > 0: --- False

10 elif -243 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(13) = '13'

1 def factorize(n=13)

2 res = ''

3 while 13 > 2 and 13 % 2 == 0: --- False

6 d = 3

7 while 13 > 3: --- True

8 if 13 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 13 > 5: --- True

8 if 13 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 13 > 7: --- True

8 if 13 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 13 > 9: --- True

8 if 13 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 13 > 11: --- True

8 if 13 % 11 == 0: --- False

11 else:

12 d = 11 + 2

d = 13

7 while 13 > 13: --- False

13 return '' + str(13)

return '13'

7. remove\_digit(71, 1) = 7

1 def remove\_digit(number=71, digit=1)

2 res = 0

3 power = 1

4 while 71 > 0: --- True

5 cur\_digit = 71 % 10

cur\_digit = 1

6 if 1 != 1: --- False

9 number = 71 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 1: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 7

Вариант: 1-1-17

1. gcd(20, 65) = 5

1 def gcd(x=20, y=65)

2 if 20 < 0: --- False

4 if 65 < 0: --- False

6 while 65 != 0: --- True

7 rem = 20 % 65

rem = 20

8 x = 65

9 y = 20

6 while 20 != 0: --- True

7 rem = 65 % 20

rem = 5

8 x = 20

9 y = 5

6 while 5 != 0: --- True

7 rem = 20 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, 26) = 26

1 def gcd(x=0, y=26)

2 if 0 < 0: --- False

4 if 26 < 0: --- False

6 while 26 != 0: --- True

7 rem = 0 % 26

rem = 0

8 x = 26

9 y = 0

6 while 0 != 0: --- False

10 return 26

3. hex(221) = 'DD'

1 def hex(number=221)

2 if 221 == 0: --- False

4 res = ''

5 while 221 > 0: --- True

6 digit = 221 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + ''

res = 'D'

24 number = 221 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + 'D'

res = 'DD'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DD'

4. square\_equal(10, 6, -28) = '-2.0 and 1.4'

3 def square\_equal(a=10, b=6, c=-28)

4 if 10 != 0: --- True

5 D = 6\*6 - 4\*10\*-28

D = 1156

6 if 1156 > 0: --- True

7 x1 = (-6 - sqrt(1156)) / (2\*10)

x1 = -2.0

8 x2 = (-6 + sqrt(1156)) / (2\*10)

x2 = 1.4

9 return str(-2.0) + ' and ' + str(1.4)

return '-2.0 and 1.4'

5. square\_equal(93, -28, 4) = 'no roots'

3 def square\_equal(a=93, b=-28, c=4)

4 if 93 != 0: --- True

5 D = -28\*-28 - 4\*93\*4

D = -704

6 if -704 > 0: --- False

10 elif -704 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(225) = '3\*3\*5\*5'

1 def factorize(n=225)

2 res = ''

3 while 225 > 2 and 225 % 2 == 0: --- False

6 d = 3

7 while 225 > 3: --- True

8 if 225 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 225 // 3

n = 75

7 while 75 > 3: --- True

8 if 75 % 3 == 0: --- True

9 res = '3\*' + str(3) + '\*'

res = '3\*3\*'

10 n = 75 // 3

n = 25

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '3\*3\*' + str(5) + '\*'

res = '3\*3\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '3\*3\*5\*' + str(5)

return '3\*3\*5\*5'

7. remove\_digit(224, 4) = 22

1 def remove\_digit(number=224, digit=4)

2 res = 0

3 power = 1

4 while 224 > 0: --- True

5 cur\_digit = 224 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 224 // 10

number = 22

4 while 22 > 0: --- True

5 cur\_digit = 22 % 10

cur\_digit = 2

6 if 2 != 4: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 22 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 4: --- True

7 res = 2 + 2 \* 10

res = 22

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 22

Вариант: 1-1-18

1. gcd(-55, -15) = 5

1 def gcd(x=-55, y=-15)

2 if -55 < 0: --- True

3 x = --55

x = 55

4 if -15 < 0: --- True

5 y = --15

y = 15

6 while 15 != 0: --- True

7 rem = 55 % 15

rem = 10

8 x = 15

9 y = 10

6 while 10 != 0: --- True

7 rem = 15 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(31, 0) = 31

1 def gcd(x=31, y=0)

2 if 31 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 31

3. hex(168) = 'A8'

1 def hex(number=168)

2 if 168 == 0: --- False

4 res = ''

5 while 168 > 0: --- True

6 digit = 168 % 16

digit = 8

7 if 8 <= 9: --- True

8 digit = str(8)

digit = '8'

23 res = '8' + ''

res = '8'

24 number = 168 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '8'

res = 'A8'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A8'

4. square\_equal(-25, -85, -60) = '-1.0 and -2.4'

3 def square\_equal(a=-25, b=-85, c=-60)

4 if -25 != 0: --- True

5 D = -85\*-85 - 4\*-25\*-60

D = 1225

6 if 1225 > 0: --- True

7 x1 = (--85 - sqrt(1225)) / (2\*-25)

x1 = -1.0

8 x2 = (--85 + sqrt(1225)) / (2\*-25)

x2 = -2.4

9 return str(-1.0) + ' and ' + str(-2.4)

return '-1.0 and -2.4'

5. square\_equal(-59, 63, -54) = 'no roots'

3 def square\_equal(a=-59, b=63, c=-54)

4 if -59 != 0: --- True

5 D = 63\*63 - 4\*-59\*-54

D = -8775

6 if -8775 > 0: --- False

10 elif -8775 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(200) = '2\*2\*2\*5\*5'

1 def factorize(n=200)

2 res = ''

3 while 200 > 2 and 200 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 200 // 2

n = 100

3 while 100 > 2 and 100 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 100 // 2

n = 50

3 while 50 > 2 and 50 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 50 // 2

n = 25

3 while 25 > 2 and 25 % 2 == 0: --- False

6 d = 3

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '2\*2\*2\*' + str(5) + '\*'

res = '2\*2\*2\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '2\*2\*2\*5\*' + str(5)

return '2\*2\*2\*5\*5'

7. remove\_digit(255, 5) = 2

1 def remove\_digit(number=255, digit=5)

2 res = 0

3 power = 1

4 while 255 > 0: --- True

5 cur\_digit = 255 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 255 // 10

number = 25

4 while 25 > 0: --- True

5 cur\_digit = 25 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 25 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 5: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 2

Вариант: 1-1-19

1. gcd(99, 21) = 3

1 def gcd(x=99, y=21)

2 if 99 < 0: --- False

4 if 21 < 0: --- False

6 while 21 != 0: --- True

7 rem = 99 % 21

rem = 15

8 x = 21

9 y = 15

6 while 15 != 0: --- True

7 rem = 21 % 15

rem = 6

8 x = 15

9 y = 6

6 while 6 != 0: --- True

7 rem = 15 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, -87) = 87

1 def gcd(x=0, y=-87)

2 if 0 < 0: --- False

4 if -87 < 0: --- True

5 y = --87

y = 87

6 while 87 != 0: --- True

7 rem = 0 % 87

rem = 0

8 x = 87

9 y = 0

6 while 0 != 0: --- False

10 return 87

3. hex(192) = 'C0'

1 def hex(number=192)

2 if 192 == 0: --- False

4 res = ''

5 while 192 > 0: --- True

6 digit = 192 % 16

digit = 0

7 if 0 <= 9: --- True

8 digit = str(0)

digit = '0'

23 res = '0' + ''

res = '0'

24 number = 192 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '0'

res = 'C0'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C0'

4. square\_equal(-2, 33, 74) = '18.5 and -2.0'

3 def square\_equal(a=-2, b=33, c=74)

4 if -2 != 0: --- True

5 D = 33\*33 - 4\*-2\*74

D = 1681

6 if 1681 > 0: --- True

7 x1 = (-33 - sqrt(1681)) / (2\*-2)

x1 = 18.5

8 x2 = (-33 + sqrt(1681)) / (2\*-2)

x2 = -2.0

9 return str(18.5) + ' and ' + str(-2.0)

return '18.5 and -2.0'

5. square\_equal(28, 81, 91) = 'no roots'

3 def square\_equal(a=28, b=81, c=91)

4 if 28 != 0: --- True

5 D = 81\*81 - 4\*28\*91

D = -3631

6 if -3631 > 0: --- False

10 elif -3631 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(11) = '11'

1 def factorize(n=11)

2 res = ''

3 while 11 > 2 and 11 % 2 == 0: --- False

6 d = 3

7 while 11 > 3: --- True

8 if 11 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 11 > 5: --- True

8 if 11 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 11 > 7: --- True

8 if 11 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 11 > 9: --- True

8 if 11 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 11 > 11: --- False

13 return '' + str(11)

return '11'

7. remove\_digit(733, 3) = 7

1 def remove\_digit(number=733, digit=3)

2 res = 0

3 power = 1

4 while 733 > 0: --- True

5 cur\_digit = 733 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 733 // 10

number = 73

4 while 73 > 0: --- True

5 cur\_digit = 73 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 73 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 3: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 7

Вариант: 1-1-20

1. gcd(-96, 57) = 3

1 def gcd(x=-96, y=57)

2 if -96 < 0: --- True

3 x = --96

x = 96

4 if 57 < 0: --- False

6 while 57 != 0: --- True

7 rem = 96 % 57

rem = 39

8 x = 57

9 y = 39

6 while 39 != 0: --- True

7 rem = 57 % 39

rem = 18

8 x = 39

9 y = 18

6 while 18 != 0: --- True

7 rem = 39 % 18

rem = 3

8 x = 18

9 y = 3

6 while 3 != 0: --- True

7 rem = 18 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, -64) = 64

1 def gcd(x=0, y=-64)

2 if 0 < 0: --- False

4 if -64 < 0: --- True

5 y = --64

y = 64

6 while 64 != 0: --- True

7 rem = 0 % 64

rem = 0

8 x = 64

9 y = 0

6 while 0 != 0: --- False

10 return 64

3. hex(205) = 'CD'

1 def hex(number=205)

2 if 205 == 0: --- False

4 res = ''

5 while 205 > 0: --- True

6 digit = 205 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + ''

res = 'D'

24 number = 205 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + 'D'

res = 'CD'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'CD'

4. square\_equal(-4, 50, -24) = '12.0 and 0.5'

3 def square\_equal(a=-4, b=50, c=-24)

4 if -4 != 0: --- True

5 D = 50\*50 - 4\*-4\*-24

D = 2116

6 if 2116 > 0: --- True

7 x1 = (-50 - sqrt(2116)) / (2\*-4)

x1 = 12.0

8 x2 = (-50 + sqrt(2116)) / (2\*-4)

x2 = 0.5

9 return str(12.0) + ' and ' + str(0.5)

return '12.0 and 0.5'

5. square\_equal(-71, 90, -57) = 'no roots'

3 def square\_equal(a=-71, b=90, c=-57)

4 if -71 != 0: --- True

5 D = 90\*90 - 4\*-71\*-57

D = -8088

6 if -8088 > 0: --- False

10 elif -8088 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(48) = '2\*2\*2\*2\*3'

1 def factorize(n=48)

2 res = ''

3 while 48 > 2 and 48 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 48 // 2

n = 24

3 while 24 > 2 and 24 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 24 // 2

n = 12

3 while 12 > 2 and 12 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 12 // 2

n = 6

3 while 6 > 2 and 6 % 2 == 0: --- True

4 res = '2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*'

5 n = 6 // 2

n = 3

3 while 3 > 2 and 3 % 2 == 0: --- False

6 d = 3

7 while 3 > 3: --- False

13 return '2\*2\*2\*2\*' + str(3)

return '2\*2\*2\*2\*3'

7. remove\_digit(891, 9) = 81

1 def remove\_digit(number=891, digit=9)

2 res = 0

3 power = 1

4 while 891 > 0: --- True

5 cur\_digit = 891 % 10

cur\_digit = 1

6 if 1 != 9: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 891 // 10

number = 89

4 while 89 > 0: --- True

5 cur\_digit = 89 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 89 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 9: --- True

7 res = 1 + 8 \* 10

res = 81

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 81

Вариант: 1-1-21

1. gcd(-42, -78) = 6

1 def gcd(x=-42, y=-78)

2 if -42 < 0: --- True

3 x = --42

x = 42

4 if -78 < 0: --- True

5 y = --78

y = 78

6 while 78 != 0: --- True

7 rem = 42 % 78

rem = 42

8 x = 78

9 y = 42

6 while 42 != 0: --- True

7 rem = 78 % 42

rem = 36

8 x = 42

9 y = 36

6 while 36 != 0: --- True

7 rem = 42 % 36

rem = 6

8 x = 36

9 y = 6

6 while 6 != 0: --- True

7 rem = 36 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(0, 58) = 58

1 def gcd(x=0, y=58)

2 if 0 < 0: --- False

4 if 58 < 0: --- False

6 while 58 != 0: --- True

7 rem = 0 % 58

rem = 0

8 x = 58

9 y = 0

6 while 0 != 0: --- False

10 return 58

3. hex(245) = 'F5'

1 def hex(number=245)

2 if 245 == 0: --- False

4 res = ''

5 while 245 > 0: --- True

6 digit = 245 % 16

digit = 5

7 if 5 <= 9: --- True

8 digit = str(5)

digit = '5'

23 res = '5' + ''

res = '5'

24 number = 245 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '5'

res = 'F5'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F5'

4. square\_equal(-15, 57, -54) = '2.0 and 1.8'

3 def square\_equal(a=-15, b=57, c=-54)

4 if -15 != 0: --- True

5 D = 57\*57 - 4\*-15\*-54

D = 9

6 if 9 > 0: --- True

7 x1 = (-57 - sqrt(9)) / (2\*-15)

x1 = 2.0

8 x2 = (-57 + sqrt(9)) / (2\*-15)

x2 = 1.8

9 return str(2.0) + ' and ' + str(1.8)

return '2.0 and 1.8'

5. square\_equal(-34, -41, -32) = 'no roots'

3 def square\_equal(a=-34, b=-41, c=-32)

4 if -34 != 0: --- True

5 D = -41\*-41 - 4\*-34\*-32

D = -2671

6 if -2671 > 0: --- False

10 elif -2671 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(7) = '7'

1 def factorize(n=7)

2 res = ''

3 while 7 > 2 and 7 % 2 == 0: --- False

6 d = 3

7 while 7 > 3: --- True

8 if 7 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '' + str(7)

return '7'

7. remove\_digit(718, 8) = 71

1 def remove\_digit(number=718, digit=8)

2 res = 0

3 power = 1

4 while 718 > 0: --- True

5 cur\_digit = 718 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 718 // 10

number = 71

4 while 71 > 0: --- True

5 cur\_digit = 71 % 10

cur\_digit = 1

6 if 1 != 8: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 71 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 8: --- True

7 res = 1 + 7 \* 10

res = 71

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 71

Вариант: 1-1-22

1. gcd(63, 51) = 3

1 def gcd(x=63, y=51)

2 if 63 < 0: --- False

4 if 51 < 0: --- False

6 while 51 != 0: --- True

7 rem = 63 % 51

rem = 12

8 x = 51

9 y = 12

6 while 12 != 0: --- True

7 rem = 51 % 12

rem = 3

8 x = 12

9 y = 3

6 while 3 != 0: --- True

7 rem = 12 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, 77) = 77

1 def gcd(x=0, y=77)

2 if 0 < 0: --- False

4 if 77 < 0: --- False

6 while 77 != 0: --- True

7 rem = 0 % 77

rem = 0

8 x = 77

9 y = 0

6 while 0 != 0: --- False

10 return 77

3. hex(239) = 'EF'

1 def hex(number=239)

2 if 239 == 0: --- False

4 res = ''

5 while 239 > 0: --- True

6 digit = 239 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + ''

res = 'F'

24 number = 239 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + 'F'

res = 'EF'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'EF'

4. square\_equal(20, 14, -90) = '-2.5 and 1.8'

3 def square\_equal(a=20, b=14, c=-90)

4 if 20 != 0: --- True

5 D = 14\*14 - 4\*20\*-90

D = 7396

6 if 7396 > 0: --- True

7 x1 = (-14 - sqrt(7396)) / (2\*20)

x1 = -2.5

8 x2 = (-14 + sqrt(7396)) / (2\*20)

x2 = 1.8

9 return str(-2.5) + ' and ' + str(1.8)

return '-2.5 and 1.8'

5. square\_equal(67, -48, 43) = 'no roots'

3 def square\_equal(a=67, b=-48, c=43)

4 if 67 != 0: --- True

5 D = -48\*-48 - 4\*67\*43

D = -9220

6 if -9220 > 0: --- False

10 elif -9220 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(4) = '2\*2'

1 def factorize(n=4)

2 res = ''

3 while 4 > 2 and 4 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 4 // 2

n = 2

3 while 2 > 2 and 2 % 2 == 0: --- False

6 d = 3

7 while 2 > 3: --- False

13 return '2\*' + str(2)

return '2\*2'

7. remove\_digit(371, 1) = 37

1 def remove\_digit(number=371, digit=1)

2 res = 0

3 power = 1

4 while 371 > 0: --- True

5 cur\_digit = 371 % 10

cur\_digit = 1

6 if 1 != 1: --- False

9 number = 371 // 10

number = 37

4 while 37 > 0: --- True

5 cur\_digit = 37 % 10

cur\_digit = 7

6 if 7 != 1: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 37 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 1: --- True

7 res = 7 + 3 \* 10

res = 37

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 37

Вариант: 1-1-23

1. gcd(-6, 33) = 3

1 def gcd(x=-6, y=33)

2 if -6 < 0: --- True

3 x = --6

x = 6

4 if 33 < 0: --- False

6 while 33 != 0: --- True

7 rem = 6 % 33

rem = 6

8 x = 33

9 y = 6

6 while 6 != 0: --- True

7 rem = 33 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, 89) = 89

1 def gcd(x=0, y=89)

2 if 0 < 0: --- False

4 if 89 < 0: --- False

6 while 89 != 0: --- True

7 rem = 0 % 89

rem = 0

8 x = 89

9 y = 0

6 while 0 != 0: --- False

10 return 89

3. hex(206) = 'CE'

1 def hex(number=206)

2 if 206 == 0: --- False

4 res = ''

5 while 206 > 0: --- True

6 digit = 206 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + ''

res = 'E'

24 number = 206 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + 'E'

res = 'CE'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'CE'

4. square\_equal(-20, 20, 15) = '1.5 and -0.5'

3 def square\_equal(a=-20, b=20, c=15)

4 if -20 != 0: --- True

5 D = 20\*20 - 4\*-20\*15

D = 1600

6 if 1600 > 0: --- True

7 x1 = (-20 - sqrt(1600)) / (2\*-20)

x1 = 1.5

8 x2 = (-20 + sqrt(1600)) / (2\*-20)

x2 = -0.5

9 return str(1.5) + ' and ' + str(-0.5)

return '1.5 and -0.5'

5. square\_equal(-28, 29, -33) = 'no roots'

3 def square\_equal(a=-28, b=29, c=-33)

4 if -28 != 0: --- True

5 D = 29\*29 - 4\*-28\*-33

D = -2855

6 if -2855 > 0: --- False

10 elif -2855 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(20) = '2\*2\*5'

1 def factorize(n=20)

2 res = ''

3 while 20 > 2 and 20 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 20 // 2

n = 10

3 while 10 > 2 and 10 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 10 // 2

n = 5

3 while 5 > 2 and 5 % 2 == 0: --- False

6 d = 3

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*' + str(5)

return '2\*2\*5'

7. remove\_digit(381, 1) = 38

1 def remove\_digit(number=381, digit=1)

2 res = 0

3 power = 1

4 while 381 > 0: --- True

5 cur\_digit = 381 % 10

cur\_digit = 1

6 if 1 != 1: --- False

9 number = 381 // 10

number = 38

4 while 38 > 0: --- True

5 cur\_digit = 38 % 10

cur\_digit = 8

6 if 8 != 1: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 38 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 1: --- True

7 res = 8 + 3 \* 10

res = 38

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 38

Вариант: 1-1-24

1. gcd(-18, 99) = 9

1 def gcd(x=-18, y=99)

2 if -18 < 0: --- True

3 x = --18

x = 18

4 if 99 < 0: --- False

6 while 99 != 0: --- True

7 rem = 18 % 99

rem = 18

8 x = 99

9 y = 18

6 while 18 != 0: --- True

7 rem = 99 % 18

rem = 9

8 x = 18

9 y = 9

6 while 9 != 0: --- True

7 rem = 18 % 9

rem = 0

8 x = 9

9 y = 0

6 while 0 != 0: --- False

10 return 9

2. gcd(0, 70) = 70

1 def gcd(x=0, y=70)

2 if 0 < 0: --- False

4 if 70 < 0: --- False

6 while 70 != 0: --- True

7 rem = 0 % 70

rem = 0

8 x = 70

9 y = 0

6 while 0 != 0: --- False

10 return 70

3. hex(241) = 'F1'

1 def hex(number=241)

2 if 241 == 0: --- False

4 res = ''

5 while 241 > 0: --- True

6 digit = 241 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 241 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '1'

res = 'F1'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F1'

4. square\_equal(-20, -43, -6) = '-0.15 and -2.0'

3 def square\_equal(a=-20, b=-43, c=-6)

4 if -20 != 0: --- True

5 D = -43\*-43 - 4\*-20\*-6

D = 1369

6 if 1369 > 0: --- True

7 x1 = (--43 - sqrt(1369)) / (2\*-20)

x1 = -0.15

8 x2 = (--43 + sqrt(1369)) / (2\*-20)

x2 = -2.0

9 return str(-0.15) + ' and ' + str(-2.0)

return '-0.15 and -2.0'

5. square\_equal(-87, -58, -27) = 'no roots'

3 def square\_equal(a=-87, b=-58, c=-27)

4 if -87 != 0: --- True

5 D = -58\*-58 - 4\*-87\*-27

D = -6032

6 if -6032 > 0: --- False

10 elif -6032 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(135) = '3\*3\*3\*5'

1 def factorize(n=135)

2 res = ''

3 while 135 > 2 and 135 % 2 == 0: --- False

6 d = 3

7 while 135 > 3: --- True

8 if 135 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 135 // 3

n = 45

7 while 45 > 3: --- True

8 if 45 % 3 == 0: --- True

9 res = '3\*' + str(3) + '\*'

res = '3\*3\*'

10 n = 45 // 3

n = 15

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '3\*3\*' + str(3) + '\*'

res = '3\*3\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '3\*3\*3\*' + str(5)

return '3\*3\*3\*5'

7. remove\_digit(241, 4) = 21

1 def remove\_digit(number=241, digit=4)

2 res = 0

3 power = 1

4 while 241 > 0: --- True

5 cur\_digit = 241 % 10

cur\_digit = 1

6 if 1 != 4: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 241 // 10

number = 24

4 while 24 > 0: --- True

5 cur\_digit = 24 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 24 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 4: --- True

7 res = 1 + 2 \* 10

res = 21

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 21

Вариант: 1-1-25

1. gcd(-88, 24) = 8

1 def gcd(x=-88, y=24)

2 if -88 < 0: --- True

3 x = --88

x = 88

4 if 24 < 0: --- False

6 while 24 != 0: --- True

7 rem = 88 % 24

rem = 16

8 x = 24

9 y = 16

6 while 16 != 0: --- True

7 rem = 24 % 16

rem = 8

8 x = 16

9 y = 8

6 while 8 != 0: --- True

7 rem = 16 % 8

rem = 0

8 x = 8

9 y = 0

6 while 0 != 0: --- False

10 return 8

2. gcd(-35, 0) = 35

1 def gcd(x=-35, y=0)

2 if -35 < 0: --- True

3 x = --35

x = 35

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 35

3. hex(247) = 'F7'

1 def hex(number=247)

2 if 247 == 0: --- False

4 res = ''

5 while 247 > 0: --- True

6 digit = 247 % 16

digit = 7

7 if 7 <= 9: --- True

8 digit = str(7)

digit = '7'

23 res = '7' + ''

res = '7'

24 number = 247 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '7'

res = 'F7'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F7'

4. square\_equal(50, 96, 46) = '-1.0 and -0.92'

3 def square\_equal(a=50, b=96, c=46)

4 if 50 != 0: --- True

5 D = 96\*96 - 4\*50\*46

D = 16

6 if 16 > 0: --- True

7 x1 = (-96 - sqrt(16)) / (2\*50)

x1 = -1.0

8 x2 = (-96 + sqrt(16)) / (2\*50)

x2 = -0.92

9 return str(-1.0) + ' and ' + str(-0.92)

return '-1.0 and -0.92'

5. square\_equal(-49, -92, -53) = 'no roots'

3 def square\_equal(a=-49, b=-92, c=-53)

4 if -49 != 0: --- True

5 D = -92\*-92 - 4\*-49\*-53

D = -1924

6 if -1924 > 0: --- False

10 elif -1924 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(300) = '2\*2\*3\*5\*5'

1 def factorize(n=300)

2 res = ''

3 while 300 > 2 and 300 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 300 // 2

n = 150

3 while 150 > 2 and 150 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 150 // 2

n = 75

3 while 75 > 2 and 75 % 2 == 0: --- False

6 d = 3

7 while 75 > 3: --- True

8 if 75 % 3 == 0: --- True

9 res = '2\*2\*' + str(3) + '\*'

res = '2\*2\*3\*'

10 n = 75 // 3

n = 25

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '2\*2\*3\*' + str(5) + '\*'

res = '2\*2\*3\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '2\*2\*3\*5\*' + str(5)

return '2\*2\*3\*5\*5'

7. remove\_digit(427, 7) = 42

1 def remove\_digit(number=427, digit=7)

2 res = 0

3 power = 1

4 while 427 > 0: --- True

5 cur\_digit = 427 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 427 // 10

number = 42

4 while 42 > 0: --- True

5 cur\_digit = 42 % 10

cur\_digit = 2

6 if 2 != 7: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 42 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 7: --- True

7 res = 2 + 4 \* 10

res = 42

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 42

Вариант: 1-1-26

1. gcd(10, -15) = 5

1 def gcd(x=10, y=-15)

2 if 10 < 0: --- False

4 if -15 < 0: --- True

5 y = --15

y = 15

6 while 15 != 0: --- True

7 rem = 10 % 15

rem = 10

8 x = 15

9 y = 10

6 while 10 != 0: --- True

7 rem = 15 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(96, 0) = 96

1 def gcd(x=96, y=0)

2 if 96 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 96

3. hex(167) = 'A7'

1 def hex(number=167)

2 if 167 == 0: --- False

4 res = ''

5 while 167 > 0: --- True

6 digit = 167 % 16

digit = 7

7 if 7 <= 9: --- True

8 digit = str(7)

digit = '7'

23 res = '7' + ''

res = '7'

24 number = 167 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '7'

res = 'A7'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A7'

4. square\_equal(0, 50, -89) = '1.78'

3 def square\_equal(a=0, b=50, c=-89)

4 if 0 != 0: --- False

14 else:

15 if 50 != 0: --- True

16 return str(--89 / 50)

return '1.78'

5. square\_equal(100, 88, 37) = 'no roots'

3 def square\_equal(a=100, b=88, c=37)

4 if 100 != 0: --- True

5 D = 88\*88 - 4\*100\*37

D = -7056

6 if -7056 > 0: --- False

10 elif -7056 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(16) = '2\*2\*2\*2'

1 def factorize(n=16)

2 res = ''

3 while 16 > 2 and 16 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 16 // 2

n = 8

3 while 8 > 2 and 8 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 8 // 2

n = 4

3 while 4 > 2 and 4 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 4 // 2

n = 2

3 while 2 > 2 and 2 % 2 == 0: --- False

6 d = 3

7 while 2 > 3: --- False

13 return '2\*2\*2\*' + str(2)

return '2\*2\*2\*2'

7. remove\_digit(485, 5) = 48

1 def remove\_digit(number=485, digit=5)

2 res = 0

3 power = 1

4 while 485 > 0: --- True

5 cur\_digit = 485 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 485 // 10

number = 48

4 while 48 > 0: --- True

5 cur\_digit = 48 % 10

cur\_digit = 8

6 if 8 != 5: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 48 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 5: --- True

7 res = 8 + 4 \* 10

res = 48

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 48

Вариант: 1-1-27

1. gcd(93, 60) = 3

1 def gcd(x=93, y=60)

2 if 93 < 0: --- False

4 if 60 < 0: --- False

6 while 60 != 0: --- True

7 rem = 93 % 60

rem = 33

8 x = 60

9 y = 33

6 while 33 != 0: --- True

7 rem = 60 % 33

rem = 27

8 x = 33

9 y = 27

6 while 27 != 0: --- True

7 rem = 33 % 27

rem = 6

8 x = 27

9 y = 6

6 while 6 != 0: --- True

7 rem = 27 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, -89) = 89

1 def gcd(x=0, y=-89)

2 if 0 < 0: --- False

4 if -89 < 0: --- True

5 y = --89

y = 89

6 while 89 != 0: --- True

7 rem = 0 % 89

rem = 0

8 x = 89

9 y = 0

6 while 0 != 0: --- False

10 return 89

3. hex(223) = 'DF'

1 def hex(number=223)

2 if 223 == 0: --- False

4 res = ''

5 while 223 > 0: --- True

6 digit = 223 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + ''

res = 'F'

24 number = 223 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + 'F'

res = 'DF'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DF'

4. square\_equal(1, 2, -3) = '-3.0 and 1.0'

3 def square\_equal(a=1, b=2, c=-3)

4 if 1 != 0: --- True

5 D = 2\*2 - 4\*1\*-3

D = 16

6 if 16 > 0: --- True

7 x1 = (-2 - sqrt(16)) / (2\*1)

x1 = -3.0

8 x2 = (-2 + sqrt(16)) / (2\*1)

x2 = 1.0

9 return str(-3.0) + ' and ' + str(1.0)

return '-3.0 and 1.0'

5. square\_equal(29, -35, 49) = 'no roots'

3 def square\_equal(a=29, b=-35, c=49)

4 if 29 != 0: --- True

5 D = -35\*-35 - 4\*29\*49

D = -4459

6 if -4459 > 0: --- False

10 elif -4459 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(64) = '2\*2\*2\*2\*2\*2'

1 def factorize(n=64)

2 res = ''

3 while 64 > 2 and 64 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 64 // 2

n = 32

3 while 32 > 2 and 32 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 32 // 2

n = 16

3 while 16 > 2 and 16 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 16 // 2

n = 8

3 while 8 > 2 and 8 % 2 == 0: --- True

4 res = '2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*'

5 n = 8 // 2

n = 4

3 while 4 > 2 and 4 % 2 == 0: --- True

4 res = '2\*2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*2\*'

5 n = 4 // 2

n = 2

3 while 2 > 2 and 2 % 2 == 0: --- False

6 d = 3

7 while 2 > 3: --- False

13 return '2\*2\*2\*2\*2\*' + str(2)

return '2\*2\*2\*2\*2\*2'

7. remove\_digit(153, 3) = 15

1 def remove\_digit(number=153, digit=3)

2 res = 0

3 power = 1

4 while 153 > 0: --- True

5 cur\_digit = 153 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 153 // 10

number = 15

4 while 15 > 0: --- True

5 cur\_digit = 15 % 10

cur\_digit = 5

6 if 5 != 3: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 15 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 3: --- True

7 res = 5 + 1 \* 10

res = 15

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 15

Вариант: 1-1-28

1. gcd(-92, -28) = 4

1 def gcd(x=-92, y=-28)

2 if -92 < 0: --- True

3 x = --92

x = 92

4 if -28 < 0: --- True

5 y = --28

y = 28

6 while 28 != 0: --- True

7 rem = 92 % 28

rem = 8

8 x = 28

9 y = 8

6 while 8 != 0: --- True

7 rem = 28 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 86) = 86

1 def gcd(x=0, y=86)

2 if 0 < 0: --- False

4 if 86 < 0: --- False

6 while 86 != 0: --- True

7 rem = 0 % 86

rem = 0

8 x = 86

9 y = 0

6 while 0 != 0: --- False

10 return 86

3. hex(253) = 'FD'

1 def hex(number=253)

2 if 253 == 0: --- False

4 res = ''

5 while 253 > 0: --- True

6 digit = 253 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + ''

res = 'D'

24 number = 253 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + 'D'

res = 'FD'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'FD'

4. square\_equal(4, -25, 25) = '1.25 and 5.0'

3 def square\_equal(a=4, b=-25, c=25)

4 if 4 != 0: --- True

5 D = -25\*-25 - 4\*4\*25

D = 225

6 if 225 > 0: --- True

7 x1 = (--25 - sqrt(225)) / (2\*4)

x1 = 1.25

8 x2 = (--25 + sqrt(225)) / (2\*4)

x2 = 5.0

9 return str(1.25) + ' and ' + str(5.0)

return '1.25 and 5.0'

5. square\_equal(36, -30, 52) = 'no roots'

3 def square\_equal(a=36, b=-30, c=52)

4 if 36 != 0: --- True

5 D = -30\*-30 - 4\*36\*52

D = -6588

6 if -6588 > 0: --- False

10 elif -6588 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(70) = '2\*5\*7'

1 def factorize(n=70)

2 res = ''

3 while 70 > 2 and 70 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 70 // 2

n = 35

3 while 35 > 2 and 35 % 2 == 0: --- False

6 d = 3

7 while 35 > 3: --- True

8 if 35 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 35 > 5: --- True

8 if 35 % 5 == 0: --- True

9 res = '2\*' + str(5) + '\*'

res = '2\*5\*'

10 n = 35 // 5

n = 7

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '2\*5\*' + str(7)

return '2\*5\*7'

7. remove\_digit(536, 6) = 53

1 def remove\_digit(number=536, digit=6)

2 res = 0

3 power = 1

4 while 536 > 0: --- True

5 cur\_digit = 536 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 536 // 10

number = 53

4 while 53 > 0: --- True

5 cur\_digit = 53 % 10

cur\_digit = 3

6 if 3 != 6: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 53 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 6: --- True

7 res = 3 + 5 \* 10

res = 53

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 53

Вариант: 1-1-29

1. gcd(93, -69) = 3

1 def gcd(x=93, y=-69)

2 if 93 < 0: --- False

4 if -69 < 0: --- True

5 y = --69

y = 69

6 while 69 != 0: --- True

7 rem = 93 % 69

rem = 24

8 x = 69

9 y = 24

6 while 24 != 0: --- True

7 rem = 69 % 24

rem = 21

8 x = 24

9 y = 21

6 while 21 != 0: --- True

7 rem = 24 % 21

rem = 3

8 x = 21

9 y = 3

6 while 3 != 0: --- True

7 rem = 21 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(62, 0) = 62

1 def gcd(x=62, y=0)

2 if 62 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 62

3. hex(209) = 'D1'

1 def hex(number=209)

2 if 209 == 0: --- False

4 res = ''

5 while 209 > 0: --- True

6 digit = 209 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 209 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '1'

res = 'D1'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D1'

4. square\_equal(-4, 5, 51) = '4.25 and -3.0'

3 def square\_equal(a=-4, b=5, c=51)

4 if -4 != 0: --- True

5 D = 5\*5 - 4\*-4\*51

D = 841

6 if 841 > 0: --- True

7 x1 = (-5 - sqrt(841)) / (2\*-4)

x1 = 4.25

8 x2 = (-5 + sqrt(841)) / (2\*-4)

x2 = -3.0

9 return str(4.25) + ' and ' + str(-3.0)

return '4.25 and -3.0'

5. square\_equal(49, -39, 19) = 'no roots'

3 def square\_equal(a=49, b=-39, c=19)

4 if 49 != 0: --- True

5 D = -39\*-39 - 4\*49\*19

D = -2203

6 if -2203 > 0: --- False

10 elif -2203 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(15) = '3\*5'

1 def factorize(n=15)

2 res = ''

3 while 15 > 2 and 15 % 2 == 0: --- False

6 d = 3

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '3\*' + str(5)

return '3\*5'

7. remove\_digit(638, 8) = 63

1 def remove\_digit(number=638, digit=8)

2 res = 0

3 power = 1

4 while 638 > 0: --- True

5 cur\_digit = 638 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 638 // 10

number = 63

4 while 63 > 0: --- True

5 cur\_digit = 63 % 10

cur\_digit = 3

6 if 3 != 8: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 63 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 8: --- True

7 res = 3 + 6 \* 10

res = 63

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 63

Вариант: 1-1-30

1. gcd(-85, 15) = 5

1 def gcd(x=-85, y=15)

2 if -85 < 0: --- True

3 x = --85

x = 85

4 if 15 < 0: --- False

6 while 15 != 0: --- True

7 rem = 85 % 15

rem = 10

8 x = 15

9 y = 10

6 while 10 != 0: --- True

7 rem = 15 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, 44) = 44

1 def gcd(x=0, y=44)

2 if 0 < 0: --- False

4 if 44 < 0: --- False

6 while 44 != 0: --- True

7 rem = 0 % 44

rem = 0

8 x = 44

9 y = 0

6 while 0 != 0: --- False

10 return 44

3. hex(228) = 'E4'

1 def hex(number=228)

2 if 228 == 0: --- False

4 res = ''

5 while 228 > 0: --- True

6 digit = 228 % 16

digit = 4

7 if 4 <= 9: --- True

8 digit = str(4)

digit = '4'

23 res = '4' + ''

res = '4'

24 number = 228 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '4'

res = 'E4'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E4'

4. square\_equal(72, -36, 0) = '0.0 and 0.5'

3 def square\_equal(a=72, b=-36, c=0)

4 if 72 != 0: --- True

5 D = -36\*-36 - 4\*72\*0

D = 1296

6 if 1296 > 0: --- True

7 x1 = (--36 - sqrt(1296)) / (2\*72)

x1 = 0.0

8 x2 = (--36 + sqrt(1296)) / (2\*72)

x2 = 0.5

9 return str(0.0) + ' and ' + str(0.5)

return '0.0 and 0.5'

5. square\_equal(-43, 46, -49) = 'no roots'

3 def square\_equal(a=-43, b=46, c=-49)

4 if -43 != 0: --- True

5 D = 46\*46 - 4\*-43\*-49

D = -6312

6 if -6312 > 0: --- False

10 elif -6312 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(25) = '5\*5'

1 def factorize(n=25)

2 res = ''

3 while 25 > 2 and 25 % 2 == 0: --- False

6 d = 3

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '5\*' + str(5)

return '5\*5'

7. remove\_digit(978, 8) = 97

1 def remove\_digit(number=978, digit=8)

2 res = 0

3 power = 1

4 while 978 > 0: --- True

5 cur\_digit = 978 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 978 // 10

number = 97

4 while 97 > 0: --- True

5 cur\_digit = 97 % 10

cur\_digit = 7

6 if 7 != 8: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 97 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 8: --- True

7 res = 7 + 9 \* 10

res = 97

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 97

Вариант: 1-1-31

1. gcd(60, 75) = 15

1 def gcd(x=60, y=75)

2 if 60 < 0: --- False

4 if 75 < 0: --- False

6 while 75 != 0: --- True

7 rem = 60 % 75

rem = 60

8 x = 75

9 y = 60

6 while 60 != 0: --- True

7 rem = 75 % 60

rem = 15

8 x = 60

9 y = 15

6 while 15 != 0: --- True

7 rem = 60 % 15

rem = 0

8 x = 15

9 y = 0

6 while 0 != 0: --- False

10 return 15

2. gcd(21, 0) = 21

1 def gcd(x=21, y=0)

2 if 21 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 21

3. hex(233) = 'E9'

1 def hex(number=233)

2 if 233 == 0: --- False

4 res = ''

5 while 233 > 0: --- True

6 digit = 233 % 16

digit = 9

7 if 9 <= 9: --- True

8 digit = str(9)

digit = '9'

23 res = '9' + ''

res = '9'

24 number = 233 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '9'

res = 'E9'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E9'

4. square\_equal(25, 39, -64) = '-2.56 and 1.0'

3 def square\_equal(a=25, b=39, c=-64)

4 if 25 != 0: --- True

5 D = 39\*39 - 4\*25\*-64

D = 7921

6 if 7921 > 0: --- True

7 x1 = (-39 - sqrt(7921)) / (2\*25)

x1 = -2.56

8 x2 = (-39 + sqrt(7921)) / (2\*25)

x2 = 1.0

9 return str(-2.56) + ' and ' + str(1.0)

return '-2.56 and 1.0'

5. square\_equal(13, -36, 52) = 'no roots'

3 def square\_equal(a=13, b=-36, c=52)

4 if 13 != 0: --- True

5 D = -36\*-36 - 4\*13\*52

D = -1408

6 if -1408 > 0: --- False

10 elif -1408 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(18) = '2\*3\*3'

1 def factorize(n=18)

2 res = ''

3 while 18 > 2 and 18 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 18 // 2

n = 9

3 while 9 > 2 and 9 % 2 == 0: --- False

6 d = 3

7 while 9 > 3: --- True

8 if 9 % 3 == 0: --- True

9 res = '2\*' + str(3) + '\*'

res = '2\*3\*'

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

13 return '2\*3\*' + str(3)

return '2\*3\*3'

7. remove\_digit(738, 8) = 73

1 def remove\_digit(number=738, digit=8)

2 res = 0

3 power = 1

4 while 738 > 0: --- True

5 cur\_digit = 738 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 738 // 10

number = 73

4 while 73 > 0: --- True

5 cur\_digit = 73 % 10

cur\_digit = 3

6 if 3 != 8: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 73 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 8: --- True

7 res = 3 + 7 \* 10

res = 73

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 73

Вариант: 1-1-32

1. gcd(-88, -76) = 4

1 def gcd(x=-88, y=-76)

2 if -88 < 0: --- True

3 x = --88

x = 88

4 if -76 < 0: --- True

5 y = --76

y = 76

6 while 76 != 0: --- True

7 rem = 88 % 76

rem = 12

8 x = 76

9 y = 12

6 while 12 != 0: --- True

7 rem = 76 % 12

rem = 4

8 x = 12

9 y = 4

6 while 4 != 0: --- True

7 rem = 12 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(4, 0) = 4

1 def gcd(x=4, y=0)

2 if 4 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 4

3. hex(178) = 'B2'

1 def hex(number=178)

2 if 178 == 0: --- False

4 res = ''

5 while 178 > 0: --- True

6 digit = 178 % 16

digit = 2

7 if 2 <= 9: --- True

8 digit = str(2)

digit = '2'

23 res = '2' + ''

res = '2'

24 number = 178 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + '2'

res = 'B2'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B2'

4. square\_equal(4, -74, -38) = '-0.5 and 19.0'

3 def square\_equal(a=4, b=-74, c=-38)

4 if 4 != 0: --- True

5 D = -74\*-74 - 4\*4\*-38

D = 6084

6 if 6084 > 0: --- True

7 x1 = (--74 - sqrt(6084)) / (2\*4)

x1 = -0.5

8 x2 = (--74 + sqrt(6084)) / (2\*4)

x2 = 19.0

9 return str(-0.5) + ' and ' + str(19.0)

return '-0.5 and 19.0'

5. square\_equal(70, -2, 23) = 'no roots'

3 def square\_equal(a=70, b=-2, c=23)

4 if 70 != 0: --- True

5 D = -2\*-2 - 4\*70\*23

D = -6436

6 if -6436 > 0: --- False

10 elif -6436 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(140) = '2\*2\*5\*7'

1 def factorize(n=140)

2 res = ''

3 while 140 > 2 and 140 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 140 // 2

n = 70

3 while 70 > 2 and 70 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 70 // 2

n = 35

3 while 35 > 2 and 35 % 2 == 0: --- False

6 d = 3

7 while 35 > 3: --- True

8 if 35 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 35 > 5: --- True

8 if 35 % 5 == 0: --- True

9 res = '2\*2\*' + str(5) + '\*'

res = '2\*2\*5\*'

10 n = 35 // 5

n = 7

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '2\*2\*5\*' + str(7)

return '2\*2\*5\*7'

7. remove\_digit(547, 4) = 57

1 def remove\_digit(number=547, digit=4)

2 res = 0

3 power = 1

4 while 547 > 0: --- True

5 cur\_digit = 547 % 10

cur\_digit = 7

6 if 7 != 4: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 547 // 10

number = 54

4 while 54 > 0: --- True

5 cur\_digit = 54 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 54 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 4: --- True

7 res = 7 + 5 \* 10

res = 57

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 57

Вариант: 1-1-33

1. gcd(95, 75) = 5

1 def gcd(x=95, y=75)

2 if 95 < 0: --- False

4 if 75 < 0: --- False

6 while 75 != 0: --- True

7 rem = 95 % 75

rem = 20

8 x = 75

9 y = 20

6 while 20 != 0: --- True

7 rem = 75 % 20

rem = 15

8 x = 20

9 y = 15

6 while 15 != 0: --- True

7 rem = 20 % 15

rem = 5

8 x = 15

9 y = 5

6 while 5 != 0: --- True

7 rem = 15 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(-6, 0) = 6

1 def gcd(x=-6, y=0)

2 if -6 < 0: --- True

3 x = --6

x = 6

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 6

3. hex(165) = 'A5'

1 def hex(number=165)

2 if 165 == 0: --- False

4 res = ''

5 while 165 > 0: --- True

6 digit = 165 % 16

digit = 5

7 if 5 <= 9: --- True

8 digit = str(5)

digit = '5'

23 res = '5' + ''

res = '5'

24 number = 165 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '5'

res = 'A5'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A5'

4. square\_equal(-4, -35, -24) = '-0.75 and -8.0'

3 def square\_equal(a=-4, b=-35, c=-24)

4 if -4 != 0: --- True

5 D = -35\*-35 - 4\*-4\*-24

D = 841

6 if 841 > 0: --- True

7 x1 = (--35 - sqrt(841)) / (2\*-4)

x1 = -0.75

8 x2 = (--35 + sqrt(841)) / (2\*-4)

x2 = -8.0

9 return str(-0.75) + ' and ' + str(-8.0)

return '-0.75 and -8.0'

5. square\_equal(61, 52, 18) = 'no roots'

3 def square\_equal(a=61, b=52, c=18)

4 if 61 != 0: --- True

5 D = 52\*52 - 4\*61\*18

D = -1688

6 if -1688 > 0: --- False

10 elif -1688 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(98) = '2\*7\*7'

1 def factorize(n=98)

2 res = ''

3 while 98 > 2 and 98 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 98 // 2

n = 49

3 while 49 > 2 and 49 % 2 == 0: --- False

6 d = 3

7 while 49 > 3: --- True

8 if 49 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 49 > 5: --- True

8 if 49 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '2\*' + str(7) + '\*'

res = '2\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '2\*7\*' + str(7)

return '2\*7\*7'

7. remove\_digit(7444, 4) = 7

1 def remove\_digit(number=7444, digit=4)

2 res = 0

3 power = 1

4 while 7444 > 0: --- True

5 cur\_digit = 7444 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 7444 // 10

number = 744

4 while 744 > 0: --- True

5 cur\_digit = 744 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 744 // 10

number = 74

4 while 74 > 0: --- True

5 cur\_digit = 74 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 74 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 4: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 7

Вариант: 1-1-34

1. gcd(45, -66) = 3

1 def gcd(x=45, y=-66)

2 if 45 < 0: --- False

4 if -66 < 0: --- True

5 y = --66

y = 66

6 while 66 != 0: --- True

7 rem = 45 % 66

rem = 45

8 x = 66

9 y = 45

6 while 45 != 0: --- True

7 rem = 66 % 45

rem = 21

8 x = 45

9 y = 21

6 while 21 != 0: --- True

7 rem = 45 % 21

rem = 3

8 x = 21

9 y = 3

6 while 3 != 0: --- True

7 rem = 21 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, 37) = 37

1 def gcd(x=0, y=37)

2 if 0 < 0: --- False

4 if 37 < 0: --- False

6 while 37 != 0: --- True

7 rem = 0 % 37

rem = 0

8 x = 37

9 y = 0

6 while 0 != 0: --- False

10 return 37

3. hex(172) = 'AC'

1 def hex(number=172)

2 if 172 == 0: --- False

4 res = ''

5 while 172 > 0: --- True

6 digit = 172 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + ''

res = 'C'

24 number = 172 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + 'C'

res = 'AC'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'AC'

4. square\_equal(0, 10, -54) = '5.4'

3 def square\_equal(a=0, b=10, c=-54)

4 if 0 != 0: --- False

14 else:

15 if 10 != 0: --- True

16 return str(--54 / 10)

return '5.4'

5. square\_equal(64, -10, 25) = 'no roots'

3 def square\_equal(a=64, b=-10, c=25)

4 if 64 != 0: --- True

5 D = -10\*-10 - 4\*64\*25

D = -6300

6 if -6300 > 0: --- False

10 elif -6300 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(96) = '2\*2\*2\*2\*2\*3'

1 def factorize(n=96)

2 res = ''

3 while 96 > 2 and 96 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 96 // 2

n = 48

3 while 48 > 2 and 48 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 48 // 2

n = 24

3 while 24 > 2 and 24 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 24 // 2

n = 12

3 while 12 > 2 and 12 % 2 == 0: --- True

4 res = '2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*'

5 n = 12 // 2

n = 6

3 while 6 > 2 and 6 % 2 == 0: --- True

4 res = '2\*2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*2\*'

5 n = 6 // 2

n = 3

3 while 3 > 2 and 3 % 2 == 0: --- False

6 d = 3

7 while 3 > 3: --- False

13 return '2\*2\*2\*2\*2\*' + str(3)

return '2\*2\*2\*2\*2\*3'

7. remove\_digit(148, 8) = 14

1 def remove\_digit(number=148, digit=8)

2 res = 0

3 power = 1

4 while 148 > 0: --- True

5 cur\_digit = 148 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 148 // 10

number = 14

4 while 14 > 0: --- True

5 cur\_digit = 14 % 10

cur\_digit = 4

6 if 4 != 8: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 14 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 8: --- True

7 res = 4 + 1 \* 10

res = 14

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 14

Вариант: 1-1-35

1. gcd(-16, -72) = 8

1 def gcd(x=-16, y=-72)

2 if -16 < 0: --- True

3 x = --16

x = 16

4 if -72 < 0: --- True

5 y = --72

y = 72

6 while 72 != 0: --- True

7 rem = 16 % 72

rem = 16

8 x = 72

9 y = 16

6 while 16 != 0: --- True

7 rem = 72 % 16

rem = 8

8 x = 16

9 y = 8

6 while 8 != 0: --- True

7 rem = 16 % 8

rem = 0

8 x = 8

9 y = 0

6 while 0 != 0: --- False

10 return 8

2. gcd(64, 0) = 64

1 def gcd(x=64, y=0)

2 if 64 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 64

3. hex(231) = 'E7'

1 def hex(number=231)

2 if 231 == 0: --- False

4 res = ''

5 while 231 > 0: --- True

6 digit = 231 % 16

digit = 7

7 if 7 <= 9: --- True

8 digit = str(7)

digit = '7'

23 res = '7' + ''

res = '7'

24 number = 231 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '7'

res = 'E7'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E7'

4. square\_equal(-28, 28, 56) = '2.0 and -1.0'

3 def square\_equal(a=-28, b=28, c=56)

4 if -28 != 0: --- True

5 D = 28\*28 - 4\*-28\*56

D = 7056

6 if 7056 > 0: --- True

7 x1 = (-28 - sqrt(7056)) / (2\*-28)

x1 = 2.0

8 x2 = (-28 + sqrt(7056)) / (2\*-28)

x2 = -1.0

9 return str(2.0) + ' and ' + str(-1.0)

return '2.0 and -1.0'

5. square\_equal(92, -74, 26) = 'no roots'

3 def square\_equal(a=92, b=-74, c=26)

4 if 92 != 0: --- True

5 D = -74\*-74 - 4\*92\*26

D = -4092

6 if -4092 > 0: --- False

10 elif -4092 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(54) = '2\*3\*3\*3'

1 def factorize(n=54)

2 res = ''

3 while 54 > 2 and 54 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 54 // 2

n = 27

3 while 27 > 2 and 27 % 2 == 0: --- False

6 d = 3

7 while 27 > 3: --- True

8 if 27 % 3 == 0: --- True

9 res = '2\*' + str(3) + '\*'

res = '2\*3\*'

10 n = 27 // 3

n = 9

7 while 9 > 3: --- True

8 if 9 % 3 == 0: --- True

9 res = '2\*3\*' + str(3) + '\*'

res = '2\*3\*3\*'

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

13 return '2\*3\*3\*' + str(3)

return '2\*3\*3\*3'

7. remove\_digit(136, 3) = 16

1 def remove\_digit(number=136, digit=3)

2 res = 0

3 power = 1

4 while 136 > 0: --- True

5 cur\_digit = 136 % 10

cur\_digit = 6

6 if 6 != 3: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 136 // 10

number = 13

4 while 13 > 0: --- True

5 cur\_digit = 13 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 13 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 3: --- True

7 res = 6 + 1 \* 10

res = 16

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 16

Вариант: 1-1-36

1. gcd(-9, 24) = 3

1 def gcd(x=-9, y=24)

2 if -9 < 0: --- True

3 x = --9

x = 9

4 if 24 < 0: --- False

6 while 24 != 0: --- True

7 rem = 9 % 24

rem = 9

8 x = 24

9 y = 9

6 while 9 != 0: --- True

7 rem = 24 % 9

rem = 6

8 x = 9

9 y = 6

6 while 6 != 0: --- True

7 rem = 9 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, 76) = 76

1 def gcd(x=0, y=76)

2 if 0 < 0: --- False

4 if 76 < 0: --- False

6 while 76 != 0: --- True

7 rem = 0 % 76

rem = 0

8 x = 76

9 y = 0

6 while 0 != 0: --- False

10 return 76

3. hex(227) = 'E3'

1 def hex(number=227)

2 if 227 == 0: --- False

4 res = ''

5 while 227 > 0: --- True

6 digit = 227 % 16

digit = 3

7 if 3 <= 9: --- True

8 digit = str(3)

digit = '3'

23 res = '3' + ''

res = '3'

24 number = 227 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '3'

res = 'E3'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E3'

4. square\_equal(4, 91, 87) = '-21.75 and -1.0'

3 def square\_equal(a=4, b=91, c=87)

4 if 4 != 0: --- True

5 D = 91\*91 - 4\*4\*87

D = 6889

6 if 6889 > 0: --- True

7 x1 = (-91 - sqrt(6889)) / (2\*4)

x1 = -21.75

8 x2 = (-91 + sqrt(6889)) / (2\*4)

x2 = -1.0

9 return str(-21.75) + ' and ' + str(-1.0)

return '-21.75 and -1.0'

5. square\_equal(-21, -32, -96) = 'no roots'

3 def square\_equal(a=-21, b=-32, c=-96)

4 if -21 != 0: --- True

5 D = -32\*-32 - 4\*-21\*-96

D = -7040

6 if -7040 > 0: --- False

10 elif -7040 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(28) = '2\*2\*7'

1 def factorize(n=28)

2 res = ''

3 while 28 > 2 and 28 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 28 // 2

n = 14

3 while 14 > 2 and 14 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 14 // 2

n = 7

3 while 7 > 2 and 7 % 2 == 0: --- False

6 d = 3

7 while 7 > 3: --- True

8 if 7 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '2\*2\*' + str(7)

return '2\*2\*7'

7. remove\_digit(466, 6) = 4

1 def remove\_digit(number=466, digit=6)

2 res = 0

3 power = 1

4 while 466 > 0: --- True

5 cur\_digit = 466 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 466 // 10

number = 46

4 while 46 > 0: --- True

5 cur\_digit = 46 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 46 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 6: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 4

Вариант: 1-1-37

1. gcd(68, 100) = 4

1 def gcd(x=68, y=100)

2 if 68 < 0: --- False

4 if 100 < 0: --- False

6 while 100 != 0: --- True

7 rem = 68 % 100

rem = 68

8 x = 100

9 y = 68

6 while 68 != 0: --- True

7 rem = 100 % 68

rem = 32

8 x = 68

9 y = 32

6 while 32 != 0: --- True

7 rem = 68 % 32

rem = 4

8 x = 32

9 y = 4

6 while 4 != 0: --- True

7 rem = 32 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -88) = 88

1 def gcd(x=0, y=-88)

2 if 0 < 0: --- False

4 if -88 < 0: --- True

5 y = --88

y = 88

6 while 88 != 0: --- True

7 rem = 0 % 88

rem = 0

8 x = 88

9 y = 0

6 while 0 != 0: --- False

10 return 88

3. hex(195) = 'C3'

1 def hex(number=195)

2 if 195 == 0: --- False

4 res = ''

5 while 195 > 0: --- True

6 digit = 195 % 16

digit = 3

7 if 3 <= 9: --- True

8 digit = str(3)

digit = '3'

23 res = '3' + ''

res = '3'

24 number = 195 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '3'

res = 'C3'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C3'

4. square\_equal(16, -68, 30) = '0.5 and 3.75'

3 def square\_equal(a=16, b=-68, c=30)

4 if 16 != 0: --- True

5 D = -68\*-68 - 4\*16\*30

D = 2704

6 if 2704 > 0: --- True

7 x1 = (--68 - sqrt(2704)) / (2\*16)

x1 = 0.5

8 x2 = (--68 + sqrt(2704)) / (2\*16)

x2 = 3.75

9 return str(0.5) + ' and ' + str(3.75)

return '0.5 and 3.75'

5. square\_equal(34, -33, 74) = 'no roots'

3 def square\_equal(a=34, b=-33, c=74)

4 if 34 != 0: --- True

5 D = -33\*-33 - 4\*34\*74

D = -8975

6 if -8975 > 0: --- False

10 elif -8975 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(56) = '2\*2\*2\*7'

1 def factorize(n=56)

2 res = ''

3 while 56 > 2 and 56 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 56 // 2

n = 28

3 while 28 > 2 and 28 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 28 // 2

n = 14

3 while 14 > 2 and 14 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 14 // 2

n = 7

3 while 7 > 2 and 7 % 2 == 0: --- False

6 d = 3

7 while 7 > 3: --- True

8 if 7 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '2\*2\*2\*' + str(7)

return '2\*2\*2\*7'

7. remove\_digit(102, 2) = 10

1 def remove\_digit(number=102, digit=2)

2 res = 0

3 power = 1

4 while 102 > 0: --- True

5 cur\_digit = 102 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 102 // 10

number = 10

4 while 10 > 0: --- True

5 cur\_digit = 10 % 10

cur\_digit = 0

6 if 0 != 2: --- True

7 res = 0 + 0 \* 1

res = 0

8 power = 1 \* 10

power = 10

9 number = 10 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 2: --- True

7 res = 0 + 1 \* 10

res = 10

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 10

Вариант: 1-1-38

1. gcd(-27, 45) = 9

1 def gcd(x=-27, y=45)

2 if -27 < 0: --- True

3 x = --27

x = 27

4 if 45 < 0: --- False

6 while 45 != 0: --- True

7 rem = 27 % 45

rem = 27

8 x = 45

9 y = 27

6 while 27 != 0: --- True

7 rem = 45 % 27

rem = 18

8 x = 27

9 y = 18

6 while 18 != 0: --- True

7 rem = 27 % 18

rem = 9

8 x = 18

9 y = 9

6 while 9 != 0: --- True

7 rem = 18 % 9

rem = 0

8 x = 9

9 y = 0

6 while 0 != 0: --- False

10 return 9

2. gcd(51, 0) = 51

1 def gcd(x=51, y=0)

2 if 51 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 51

3. hex(218) = 'DA'

1 def hex(number=218)

2 if 218 == 0: --- False

4 res = ''

5 while 218 > 0: --- True

6 digit = 218 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + ''

res = 'A'

24 number = 218 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + 'A'

res = 'DA'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DA'

4. square\_equal(-40, -66, -20) = '-0.4 and -1.25'

3 def square\_equal(a=-40, b=-66, c=-20)

4 if -40 != 0: --- True

5 D = -66\*-66 - 4\*-40\*-20

D = 1156

6 if 1156 > 0: --- True

7 x1 = (--66 - sqrt(1156)) / (2\*-40)

x1 = -0.4

8 x2 = (--66 + sqrt(1156)) / (2\*-40)

x2 = -1.25

9 return str(-0.4) + ' and ' + str(-1.25)

return '-0.4 and -1.25'

5. square\_equal(55, -90, 64) = 'no roots'

3 def square\_equal(a=55, b=-90, c=64)

4 if 55 != 0: --- True

5 D = -90\*-90 - 4\*55\*64

D = -5980

6 if -5980 > 0: --- False

10 elif -5980 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(350) = '2\*5\*5\*7'

1 def factorize(n=350)

2 res = ''

3 while 350 > 2 and 350 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 350 // 2

n = 175

3 while 175 > 2 and 175 % 2 == 0: --- False

6 d = 3

7 while 175 > 3: --- True

8 if 175 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 175 > 5: --- True

8 if 175 % 5 == 0: --- True

9 res = '2\*' + str(5) + '\*'

res = '2\*5\*'

10 n = 175 // 5

n = 35

7 while 35 > 5: --- True

8 if 35 % 5 == 0: --- True

9 res = '2\*5\*' + str(5) + '\*'

res = '2\*5\*5\*'

10 n = 35 // 5

n = 7

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '2\*5\*5\*' + str(7)

return '2\*5\*5\*7'

7. remove\_digit(447, 7) = 44

1 def remove\_digit(number=447, digit=7)

2 res = 0

3 power = 1

4 while 447 > 0: --- True

5 cur\_digit = 447 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 447 // 10

number = 44

4 while 44 > 0: --- True

5 cur\_digit = 44 % 10

cur\_digit = 4

6 if 4 != 7: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 44 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 7: --- True

7 res = 4 + 4 \* 10

res = 44

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 44

Вариант: 1-1-39

1. gcd(-70, -80) = 10

1 def gcd(x=-70, y=-80)

2 if -70 < 0: --- True

3 x = --70

x = 70

4 if -80 < 0: --- True

5 y = --80

y = 80

6 while 80 != 0: --- True

7 rem = 70 % 80

rem = 70

8 x = 80

9 y = 70

6 while 70 != 0: --- True

7 rem = 80 % 70

rem = 10

8 x = 70

9 y = 10

6 while 10 != 0: --- True

7 rem = 70 % 10

rem = 0

8 x = 10

9 y = 0

6 while 0 != 0: --- False

10 return 10

2. gcd(0, 55) = 55

1 def gcd(x=0, y=55)

2 if 0 < 0: --- False

4 if 55 < 0: --- False

6 while 55 != 0: --- True

7 rem = 0 % 55

rem = 0

8 x = 55

9 y = 0

6 while 0 != 0: --- False

10 return 55

3. hex(185) = 'B9'

1 def hex(number=185)

2 if 185 == 0: --- False

4 res = ''

5 while 185 > 0: --- True

6 digit = 185 % 16

digit = 9

7 if 9 <= 9: --- True

8 digit = str(9)

digit = '9'

23 res = '9' + ''

res = '9'

24 number = 185 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + '9'

res = 'B9'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B9'

4. square\_equal(-5, -16, 52) = '2.0 and -5.2'

3 def square\_equal(a=-5, b=-16, c=52)

4 if -5 != 0: --- True

5 D = -16\*-16 - 4\*-5\*52

D = 1296

6 if 1296 > 0: --- True

7 x1 = (--16 - sqrt(1296)) / (2\*-5)

x1 = 2.0

8 x2 = (--16 + sqrt(1296)) / (2\*-5)

x2 = -5.2

9 return str(2.0) + ' and ' + str(-5.2)

return '2.0 and -5.2'

5. square\_equal(-13, -28, -19) = 'no roots'

3 def square\_equal(a=-13, b=-28, c=-19)

4 if -13 != 0: --- True

5 D = -28\*-28 - 4\*-13\*-19

D = -204

6 if -204 > 0: --- False

10 elif -204 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(180) = '2\*2\*3\*3\*5'

1 def factorize(n=180)

2 res = ''

3 while 180 > 2 and 180 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 180 // 2

n = 90

3 while 90 > 2 and 90 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 90 // 2

n = 45

3 while 45 > 2 and 45 % 2 == 0: --- False

6 d = 3

7 while 45 > 3: --- True

8 if 45 % 3 == 0: --- True

9 res = '2\*2\*' + str(3) + '\*'

res = '2\*2\*3\*'

10 n = 45 // 3

n = 15

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '2\*2\*3\*' + str(3) + '\*'

res = '2\*2\*3\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*3\*3\*' + str(5)

return '2\*2\*3\*3\*5'

7. remove\_digit(264, 6) = 24

1 def remove\_digit(number=264, digit=6)

2 res = 0

3 power = 1

4 while 264 > 0: --- True

5 cur\_digit = 264 % 10

cur\_digit = 4

6 if 4 != 6: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 264 // 10

number = 26

4 while 26 > 0: --- True

5 cur\_digit = 26 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 26 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 6: --- True

7 res = 4 + 2 \* 10

res = 24

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 24

Вариант: 1-1-40

1. gcd(-44, -96) = 4

1 def gcd(x=-44, y=-96)

2 if -44 < 0: --- True

3 x = --44

x = 44

4 if -96 < 0: --- True

5 y = --96

y = 96

6 while 96 != 0: --- True

7 rem = 44 % 96

rem = 44

8 x = 96

9 y = 44

6 while 44 != 0: --- True

7 rem = 96 % 44

rem = 8

8 x = 44

9 y = 8

6 while 8 != 0: --- True

7 rem = 44 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -16) = 16

1 def gcd(x=0, y=-16)

2 if 0 < 0: --- False

4 if -16 < 0: --- True

5 y = --16

y = 16

6 while 16 != 0: --- True

7 rem = 0 % 16

rem = 0

8 x = 16

9 y = 0

6 while 0 != 0: --- False

10 return 16

3. hex(196) = 'C4'

1 def hex(number=196)

2 if 196 == 0: --- False

4 res = ''

5 while 196 > 0: --- True

6 digit = 196 % 16

digit = 4

7 if 4 <= 9: --- True

8 digit = str(4)

digit = '4'

23 res = '4' + ''

res = '4'

24 number = 196 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '4'

res = 'C4'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C4'

4. square\_equal(4, 39, 81) = '-6.75 and -3.0'

3 def square\_equal(a=4, b=39, c=81)

4 if 4 != 0: --- True

5 D = 39\*39 - 4\*4\*81

D = 225

6 if 225 > 0: --- True

7 x1 = (-39 - sqrt(225)) / (2\*4)

x1 = -6.75

8 x2 = (-39 + sqrt(225)) / (2\*4)

x2 = -3.0

9 return str(-6.75) + ' and ' + str(-3.0)

return '-6.75 and -3.0'

5. square\_equal(-20, 4, -89) = 'no roots'

3 def square\_equal(a=-20, b=4, c=-89)

4 if -20 != 0: --- True

5 D = 4\*4 - 4\*-20\*-89

D = -7104

6 if -7104 > 0: --- False

10 elif -7104 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(375) = '3\*5\*5\*5'

1 def factorize(n=375)

2 res = ''

3 while 375 > 2 and 375 % 2 == 0: --- False

6 d = 3

7 while 375 > 3: --- True

8 if 375 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 375 // 3

n = 125

7 while 125 > 3: --- True

8 if 125 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 125 > 5: --- True

8 if 125 % 5 == 0: --- True

9 res = '3\*' + str(5) + '\*'

res = '3\*5\*'

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '3\*5\*' + str(5) + '\*'

res = '3\*5\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '3\*5\*5\*' + str(5)

return '3\*5\*5\*5'

7. remove\_digit(458, 5) = 48

1 def remove\_digit(number=458, digit=5)

2 res = 0

3 power = 1

4 while 458 > 0: --- True

5 cur\_digit = 458 % 10

cur\_digit = 8

6 if 8 != 5: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 458 // 10

number = 45

4 while 45 > 0: --- True

5 cur\_digit = 45 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 45 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 5: --- True

7 res = 8 + 4 \* 10

res = 48

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 48

Вариант: 1-1-41

1. gcd(75, 65) = 5

1 def gcd(x=75, y=65)

2 if 75 < 0: --- False

4 if 65 < 0: --- False

6 while 65 != 0: --- True

7 rem = 75 % 65

rem = 10

8 x = 65

9 y = 10

6 while 10 != 0: --- True

7 rem = 65 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(77, 0) = 77

1 def gcd(x=77, y=0)

2 if 77 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 77

3. hex(225) = 'E1'

1 def hex(number=225)

2 if 225 == 0: --- False

4 res = ''

5 while 225 > 0: --- True

6 digit = 225 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 225 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '1'

res = 'E1'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E1'

4. square\_equal(4, -56, 27) = '0.5 and 13.5'

3 def square\_equal(a=4, b=-56, c=27)

4 if 4 != 0: --- True

5 D = -56\*-56 - 4\*4\*27

D = 2704

6 if 2704 > 0: --- True

7 x1 = (--56 - sqrt(2704)) / (2\*4)

x1 = 0.5

8 x2 = (--56 + sqrt(2704)) / (2\*4)

x2 = 13.5

9 return str(0.5) + ' and ' + str(13.5)

return '0.5 and 13.5'

5. square\_equal(37, 22, 30) = 'no roots'

3 def square\_equal(a=37, b=22, c=30)

4 if 37 != 0: --- True

5 D = 22\*22 - 4\*37\*30

D = -3956

6 if -3956 > 0: --- False

10 elif -3956 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(245) = '5\*7\*7'

1 def factorize(n=245)

2 res = ''

3 while 245 > 2 and 245 % 2 == 0: --- False

6 d = 3

7 while 245 > 3: --- True

8 if 245 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 245 > 5: --- True

8 if 245 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 245 // 5

n = 49

7 while 49 > 5: --- True

8 if 49 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '5\*' + str(7) + '\*'

res = '5\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '5\*7\*' + str(7)

return '5\*7\*7'

7. remove\_digit(347, 4) = 37

1 def remove\_digit(number=347, digit=4)

2 res = 0

3 power = 1

4 while 347 > 0: --- True

5 cur\_digit = 347 % 10

cur\_digit = 7

6 if 7 != 4: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 347 // 10

number = 34

4 while 34 > 0: --- True

5 cur\_digit = 34 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 34 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 4: --- True

7 res = 7 + 3 \* 10

res = 37

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 37

Вариант: 1-1-42

1. gcd(-48, 52) = 4

1 def gcd(x=-48, y=52)

2 if -48 < 0: --- True

3 x = --48

x = 48

4 if 52 < 0: --- False

6 while 52 != 0: --- True

7 rem = 48 % 52

rem = 48

8 x = 52

9 y = 48

6 while 48 != 0: --- True

7 rem = 52 % 48

rem = 4

8 x = 48

9 y = 4

6 while 4 != 0: --- True

7 rem = 48 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(-97, 0) = 97

1 def gcd(x=-97, y=0)

2 if -97 < 0: --- True

3 x = --97

x = 97

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 97

3. hex(170) = 'AA'

1 def hex(number=170)

2 if 170 == 0: --- False

4 res = ''

5 while 170 > 0: --- True

6 digit = 170 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + ''

res = 'A'

24 number = 170 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + 'A'

res = 'AA'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'AA'

4. square\_equal(-5, -10, 0) = '-0.0 and -2.0'

3 def square\_equal(a=-5, b=-10, c=0)

4 if -5 != 0: --- True

5 D = -10\*-10 - 4\*-5\*0

D = 100

6 if 100 > 0: --- True

7 x1 = (--10 - sqrt(100)) / (2\*-5)

x1 = -0.0

8 x2 = (--10 + sqrt(100)) / (2\*-5)

x2 = -2.0

9 return str(-0.0) + ' and ' + str(-2.0)

return '-0.0 and -2.0'

5. square\_equal(82, 7, 28) = 'no roots'

3 def square\_equal(a=82, b=7, c=28)

4 if 82 != 0: --- True

5 D = 7\*7 - 4\*82\*28

D = -9135

6 if -9135 > 0: --- False

10 elif -9135 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(196) = '2\*2\*7\*7'

1 def factorize(n=196)

2 res = ''

3 while 196 > 2 and 196 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 196 // 2

n = 98

3 while 98 > 2 and 98 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 98 // 2

n = 49

3 while 49 > 2 and 49 % 2 == 0: --- False

6 d = 3

7 while 49 > 3: --- True

8 if 49 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 49 > 5: --- True

8 if 49 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '2\*2\*' + str(7) + '\*'

res = '2\*2\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '2\*2\*7\*' + str(7)

return '2\*2\*7\*7'

7. remove\_digit(628, 2) = 68

1 def remove\_digit(number=628, digit=2)

2 res = 0

3 power = 1

4 while 628 > 0: --- True

5 cur\_digit = 628 % 10

cur\_digit = 8

6 if 8 != 2: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 628 // 10

number = 62

4 while 62 > 0: --- True

5 cur\_digit = 62 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 62 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 2: --- True

7 res = 8 + 6 \* 10

res = 68

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 68

Вариант: 1-1-43

1. gcd(-15, 80) = 5

1 def gcd(x=-15, y=80)

2 if -15 < 0: --- True

3 x = --15

x = 15

4 if 80 < 0: --- False

6 while 80 != 0: --- True

7 rem = 15 % 80

rem = 15

8 x = 80

9 y = 15

6 while 15 != 0: --- True

7 rem = 80 % 15

rem = 5

8 x = 15

9 y = 5

6 while 5 != 0: --- True

7 rem = 15 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, -84) = 84

1 def gcd(x=0, y=-84)

2 if 0 < 0: --- False

4 if -84 < 0: --- True

5 y = --84

y = 84

6 while 84 != 0: --- True

7 rem = 0 % 84

rem = 0

8 x = 84

9 y = 0

6 while 0 != 0: --- False

10 return 84

3. hex(191) = 'BF'

1 def hex(number=191)

2 if 191 == 0: --- False

4 res = ''

5 while 191 > 0: --- True

6 digit = 191 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + ''

res = 'F'

24 number = 191 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + 'F'

res = 'BF'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'BF'

4. square\_equal(-5, -85, 0) = '-0.0 and -17.0'

3 def square\_equal(a=-5, b=-85, c=0)

4 if -5 != 0: --- True

5 D = -85\*-85 - 4\*-5\*0

D = 7225

6 if 7225 > 0: --- True

7 x1 = (--85 - sqrt(7225)) / (2\*-5)

x1 = -0.0

8 x2 = (--85 + sqrt(7225)) / (2\*-5)

x2 = -17.0

9 return str(-0.0) + ' and ' + str(-17.0)

return '-0.0 and -17.0'

5. square\_equal(42, -23, 6) = 'no roots'

3 def square\_equal(a=42, b=-23, c=6)

4 if 42 != 0: --- True

5 D = -23\*-23 - 4\*42\*6

D = -479

6 if -479 > 0: --- False

10 elif -479 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(12) = '2\*2\*3'

1 def factorize(n=12)

2 res = ''

3 while 12 > 2 and 12 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 12 // 2

n = 6

3 while 6 > 2 and 6 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 6 // 2

n = 3

3 while 3 > 2 and 3 % 2 == 0: --- False

6 d = 3

7 while 3 > 3: --- False

13 return '2\*2\*' + str(3)

return '2\*2\*3'

7. remove\_digit(545, 4) = 55

1 def remove\_digit(number=545, digit=4)

2 res = 0

3 power = 1

4 while 545 > 0: --- True

5 cur\_digit = 545 % 10

cur\_digit = 5

6 if 5 != 4: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 545 // 10

number = 54

4 while 54 > 0: --- True

5 cur\_digit = 54 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 54 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 4: --- True

7 res = 5 + 5 \* 10

res = 55

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 55

Вариант: 1-1-44

1. gcd(-60, -51) = 3

1 def gcd(x=-60, y=-51)

2 if -60 < 0: --- True

3 x = --60

x = 60

4 if -51 < 0: --- True

5 y = --51

y = 51

6 while 51 != 0: --- True

7 rem = 60 % 51

rem = 9

8 x = 51

9 y = 9

6 while 9 != 0: --- True

7 rem = 51 % 9

rem = 6

8 x = 9

9 y = 6

6 while 6 != 0: --- True

7 rem = 9 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, -81) = 81

1 def gcd(x=0, y=-81)

2 if 0 < 0: --- False

4 if -81 < 0: --- True

5 y = --81

y = 81

6 while 81 != 0: --- True

7 rem = 0 % 81

rem = 0

8 x = 81

9 y = 0

6 while 0 != 0: --- False

10 return 81

3. hex(180) = 'B4'

1 def hex(number=180)

2 if 180 == 0: --- False

4 res = ''

5 while 180 > 0: --- True

6 digit = 180 % 16

digit = 4

7 if 4 <= 9: --- True

8 digit = str(4)

digit = '4'

23 res = '4' + ''

res = '4'

24 number = 180 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + '4'

res = 'B4'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B4'

4. square\_equal(-1, -23, 78) = '3.0 and -26.0'

3 def square\_equal(a=-1, b=-23, c=78)

4 if -1 != 0: --- True

5 D = -23\*-23 - 4\*-1\*78

D = 841

6 if 841 > 0: --- True

7 x1 = (--23 - sqrt(841)) / (2\*-1)

x1 = 3.0

8 x2 = (--23 + sqrt(841)) / (2\*-1)

x2 = -26.0

9 return str(3.0) + ' and ' + str(-26.0)

return '3.0 and -26.0'

5. square\_equal(48, 71, 32) = 'no roots'

3 def square\_equal(a=48, b=71, c=32)

4 if 48 != 0: --- True

5 D = 71\*71 - 4\*48\*32

D = -1103

6 if -1103 > 0: --- False

10 elif -1103 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(77) = '7\*11'

1 def factorize(n=77)

2 res = ''

3 while 77 > 2 and 77 % 2 == 0: --- False

6 d = 3

7 while 77 > 3: --- True

8 if 77 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 77 > 5: --- True

8 if 77 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 77 > 7: --- True

8 if 77 % 7 == 0: --- True

9 res = '' + str(7) + '\*'

res = '7\*'

10 n = 77 // 7

n = 11

7 while 11 > 7: --- True

8 if 11 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 11 > 9: --- True

8 if 11 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 11 > 11: --- False

13 return '7\*' + str(11)

return '7\*11'

7. remove\_digit(2444, 4) = 2

1 def remove\_digit(number=2444, digit=4)

2 res = 0

3 power = 1

4 while 2444 > 0: --- True

5 cur\_digit = 2444 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 2444 // 10

number = 244

4 while 244 > 0: --- True

5 cur\_digit = 244 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 244 // 10

number = 24

4 while 24 > 0: --- True

5 cur\_digit = 24 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 24 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 4: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 2

Вариант: 1-1-45

1. gcd(-24, -60) = 12

1 def gcd(x=-24, y=-60)

2 if -24 < 0: --- True

3 x = --24

x = 24

4 if -60 < 0: --- True

5 y = --60

y = 60

6 while 60 != 0: --- True

7 rem = 24 % 60

rem = 24

8 x = 60

9 y = 24

6 while 24 != 0: --- True

7 rem = 60 % 24

rem = 12

8 x = 24

9 y = 12

6 while 12 != 0: --- True

7 rem = 24 % 12

rem = 0

8 x = 12

9 y = 0

6 while 0 != 0: --- False

10 return 12

2. gcd(0, -69) = 69

1 def gcd(x=0, y=-69)

2 if 0 < 0: --- False

4 if -69 < 0: --- True

5 y = --69

y = 69

6 while 69 != 0: --- True

7 rem = 0 % 69

rem = 0

8 x = 69

9 y = 0

6 while 0 != 0: --- False

10 return 69

3. hex(187) = 'BB'

1 def hex(number=187)

2 if 187 == 0: --- False

4 res = ''

5 while 187 > 0: --- True

6 digit = 187 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + ''

res = 'B'

24 number = 187 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + 'B'

res = 'BB'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'BB'

4. square\_equal(0, -38, 57) = '1.5'

3 def square\_equal(a=0, b=-38, c=57)

4 if 0 != 0: --- False

14 else:

15 if -38 != 0: --- True

16 return str(-57 / -38)

return '1.5'

5. square\_equal(-80, -26, -17) = 'no roots'

3 def square\_equal(a=-80, b=-26, c=-17)

4 if -80 != 0: --- True

5 D = -26\*-26 - 4\*-80\*-17

D = -4764

6 if -4764 > 0: --- False

10 elif -4764 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(22) = '2\*11'

1 def factorize(n=22)

2 res = ''

3 while 22 > 2 and 22 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 22 // 2

n = 11

3 while 11 > 2 and 11 % 2 == 0: --- False

6 d = 3

7 while 11 > 3: --- True

8 if 11 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 11 > 5: --- True

8 if 11 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 11 > 7: --- True

8 if 11 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 11 > 9: --- True

8 if 11 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 11 > 11: --- False

13 return '2\*' + str(11)

return '2\*11'

7. remove\_digit(583, 8) = 53

1 def remove\_digit(number=583, digit=8)

2 res = 0

3 power = 1

4 while 583 > 0: --- True

5 cur\_digit = 583 % 10

cur\_digit = 3

6 if 3 != 8: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 583 // 10

number = 58

4 while 58 > 0: --- True

5 cur\_digit = 58 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 58 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 8: --- True

7 res = 3 + 5 \* 10

res = 53

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 53

Вариант: 1-1-46

1. gcd(-20, -65) = 5

1 def gcd(x=-20, y=-65)

2 if -20 < 0: --- True

3 x = --20

x = 20

4 if -65 < 0: --- True

5 y = --65

y = 65

6 while 65 != 0: --- True

7 rem = 20 % 65

rem = 20

8 x = 65

9 y = 20

6 while 20 != 0: --- True

7 rem = 65 % 20

rem = 5

8 x = 20

9 y = 5

6 while 5 != 0: --- True

7 rem = 20 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(1, 0) = 1

1 def gcd(x=1, y=0)

2 if 1 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 1

3. hex(211) = 'D3'

1 def hex(number=211)

2 if 211 == 0: --- False

4 res = ''

5 while 211 > 0: --- True

6 digit = 211 % 16

digit = 3

7 if 3 <= 9: --- True

8 digit = str(3)

digit = '3'

23 res = '3' + ''

res = '3'

24 number = 211 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '3'

res = 'D3'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D3'

4. square\_equal(-15, -6, 21) = '1.0 and -1.4'

3 def square\_equal(a=-15, b=-6, c=21)

4 if -15 != 0: --- True

5 D = -6\*-6 - 4\*-15\*21

D = 1296

6 if 1296 > 0: --- True

7 x1 = (--6 - sqrt(1296)) / (2\*-15)

x1 = 1.0

8 x2 = (--6 + sqrt(1296)) / (2\*-15)

x2 = -1.4

9 return str(1.0) + ' and ' + str(-1.4)

return '1.0 and -1.4'

5. square\_equal(-29, 32, -27) = 'no roots'

3 def square\_equal(a=-29, b=32, c=-27)

4 if -29 != 0: --- True

5 D = 32\*32 - 4\*-29\*-27

D = -2108

6 if -2108 > 0: --- False

10 elif -2108 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(108) = '2\*2\*3\*3\*3'

1 def factorize(n=108)

2 res = ''

3 while 108 > 2 and 108 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 108 // 2

n = 54

3 while 54 > 2 and 54 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 54 // 2

n = 27

3 while 27 > 2 and 27 % 2 == 0: --- False

6 d = 3

7 while 27 > 3: --- True

8 if 27 % 3 == 0: --- True

9 res = '2\*2\*' + str(3) + '\*'

res = '2\*2\*3\*'

10 n = 27 // 3

n = 9

7 while 9 > 3: --- True

8 if 9 % 3 == 0: --- True

9 res = '2\*2\*3\*' + str(3) + '\*'

res = '2\*2\*3\*3\*'

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

13 return '2\*2\*3\*3\*' + str(3)

return '2\*2\*3\*3\*3'

7. remove\_digit(863, 6) = 83

1 def remove\_digit(number=863, digit=6)

2 res = 0

3 power = 1

4 while 863 > 0: --- True

5 cur\_digit = 863 % 10

cur\_digit = 3

6 if 3 != 6: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 863 // 10

number = 86

4 while 86 > 0: --- True

5 cur\_digit = 86 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 86 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 6: --- True

7 res = 3 + 8 \* 10

res = 83

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 83

Вариант: 1-1-47

1. gcd(-15, 55) = 5

1 def gcd(x=-15, y=55)

2 if -15 < 0: --- True

3 x = --15

x = 15

4 if 55 < 0: --- False

6 while 55 != 0: --- True

7 rem = 15 % 55

rem = 15

8 x = 55

9 y = 15

6 while 15 != 0: --- True

7 rem = 55 % 15

rem = 10

8 x = 15

9 y = 10

6 while 10 != 0: --- True

7 rem = 15 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, -21) = 21

1 def gcd(x=0, y=-21)

2 if 0 < 0: --- False

4 if -21 < 0: --- True

5 y = --21

y = 21

6 while 21 != 0: --- True

7 rem = 0 % 21

rem = 0

8 x = 21

9 y = 0

6 while 0 != 0: --- False

10 return 21

3. hex(188) = 'BC'

1 def hex(number=188)

2 if 188 == 0: --- False

4 res = ''

5 while 188 > 0: --- True

6 digit = 188 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + ''

res = 'C'

24 number = 188 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + 'C'

res = 'BC'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'BC'

4. square\_equal(2, -36, -38) = '-1.0 and 19.0'

3 def square\_equal(a=2, b=-36, c=-38)

4 if 2 != 0: --- True

5 D = -36\*-36 - 4\*2\*-38

D = 1600

6 if 1600 > 0: --- True

7 x1 = (--36 - sqrt(1600)) / (2\*2)

x1 = -1.0

8 x2 = (--36 + sqrt(1600)) / (2\*2)

x2 = 19.0

9 return str(-1.0) + ' and ' + str(19.0)

return '-1.0 and 19.0'

5. square\_equal(-89, -62, -14) = 'no roots'

3 def square\_equal(a=-89, b=-62, c=-14)

4 if -89 != 0: --- True

5 D = -62\*-62 - 4\*-89\*-14

D = -1140

6 if -1140 > 0: --- False

10 elif -1140 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(490) = '2\*5\*7\*7'

1 def factorize(n=490)

2 res = ''

3 while 490 > 2 and 490 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 490 // 2

n = 245

3 while 245 > 2 and 245 % 2 == 0: --- False

6 d = 3

7 while 245 > 3: --- True

8 if 245 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 245 > 5: --- True

8 if 245 % 5 == 0: --- True

9 res = '2\*' + str(5) + '\*'

res = '2\*5\*'

10 n = 245 // 5

n = 49

7 while 49 > 5: --- True

8 if 49 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '2\*5\*' + str(7) + '\*'

res = '2\*5\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '2\*5\*7\*' + str(7)

return '2\*5\*7\*7'

7. remove\_digit(584, 4) = 58

1 def remove\_digit(number=584, digit=4)

2 res = 0

3 power = 1

4 while 584 > 0: --- True

5 cur\_digit = 584 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 584 // 10

number = 58

4 while 58 > 0: --- True

5 cur\_digit = 58 % 10

cur\_digit = 8

6 if 8 != 4: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 58 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 4: --- True

7 res = 8 + 5 \* 10

res = 58

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 58

Вариант: 1-1-48

1. gcd(-45, 65) = 5

1 def gcd(x=-45, y=65)

2 if -45 < 0: --- True

3 x = --45

x = 45

4 if 65 < 0: --- False

6 while 65 != 0: --- True

7 rem = 45 % 65

rem = 45

8 x = 65

9 y = 45

6 while 45 != 0: --- True

7 rem = 65 % 45

rem = 20

8 x = 45

9 y = 20

6 while 20 != 0: --- True

7 rem = 45 % 20

rem = 5

8 x = 20

9 y = 5

6 while 5 != 0: --- True

7 rem = 20 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(-20, 0) = 20

1 def gcd(x=-20, y=0)

2 if -20 < 0: --- True

3 x = --20

x = 20

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 20

3. hex(235) = 'EB'

1 def hex(number=235)

2 if 235 == 0: --- False

4 res = ''

5 while 235 > 0: --- True

6 digit = 235 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + ''

res = 'B'

24 number = 235 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + 'B'

res = 'EB'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'EB'

4. square\_equal(-5, 6, 0) = '1.2 and -0.0'

3 def square\_equal(a=-5, b=6, c=0)

4 if -5 != 0: --- True

5 D = 6\*6 - 4\*-5\*0

D = 36

6 if 36 > 0: --- True

7 x1 = (-6 - sqrt(36)) / (2\*-5)

x1 = 1.2

8 x2 = (-6 + sqrt(36)) / (2\*-5)

x2 = -0.0

9 return str(1.2) + ' and ' + str(-0.0)

return '1.2 and -0.0'

5. square\_equal(19, -4, 60) = 'no roots'

3 def square\_equal(a=19, b=-4, c=60)

4 if 19 != 0: --- True

5 D = -4\*-4 - 4\*19\*60

D = -4544

6 if -4544 > 0: --- False

10 elif -4544 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(45) = '3\*3\*5'

1 def factorize(n=45)

2 res = ''

3 while 45 > 2 and 45 % 2 == 0: --- False

6 d = 3

7 while 45 > 3: --- True

8 if 45 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 45 // 3

n = 15

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '3\*' + str(3) + '\*'

res = '3\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '3\*3\*' + str(5)

return '3\*3\*5'

7. remove\_digit(534, 3) = 54

1 def remove\_digit(number=534, digit=3)

2 res = 0

3 power = 1

4 while 534 > 0: --- True

5 cur\_digit = 534 % 10

cur\_digit = 4

6 if 4 != 3: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 534 // 10

number = 53

4 while 53 > 0: --- True

5 cur\_digit = 53 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 53 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 3: --- True

7 res = 4 + 5 \* 10

res = 54

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 54

Вариант: 1-1-49

1. gcd(-64, 80) = 16

1 def gcd(x=-64, y=80)

2 if -64 < 0: --- True

3 x = --64

x = 64

4 if 80 < 0: --- False

6 while 80 != 0: --- True

7 rem = 64 % 80

rem = 64

8 x = 80

9 y = 64

6 while 64 != 0: --- True

7 rem = 80 % 64

rem = 16

8 x = 64

9 y = 16

6 while 16 != 0: --- True

7 rem = 64 % 16

rem = 0

8 x = 16

9 y = 0

6 while 0 != 0: --- False

10 return 16

2. gcd(-55, 0) = 55

1 def gcd(x=-55, y=0)

2 if -55 < 0: --- True

3 x = --55

x = 55

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 55

3. hex(176) = 'B0'

1 def hex(number=176)

2 if 176 == 0: --- False

4 res = ''

5 while 176 > 0: --- True

6 digit = 176 % 16

digit = 0

7 if 0 <= 9: --- True

8 digit = str(0)

digit = '0'

23 res = '0' + ''

res = '0'

24 number = 176 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + '0'

res = 'B0'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B0'

4. square\_equal(0, 1, -44) = '44.0'

3 def square\_equal(a=0, b=1, c=-44)

4 if 0 != 0: --- False

14 else:

15 if 1 != 0: --- True

16 return str(--44 / 1)

return '44.0'

5. square\_equal(-79, -93, -38) = 'no roots'

3 def square\_equal(a=-79, b=-93, c=-38)

4 if -79 != 0: --- True

5 D = -93\*-93 - 4\*-79\*-38

D = -3359

6 if -3359 > 0: --- False

10 elif -3359 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(125) = '5\*5\*5'

1 def factorize(n=125)

2 res = ''

3 while 125 > 2 and 125 % 2 == 0: --- False

6 d = 3

7 while 125 > 3: --- True

8 if 125 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 125 > 5: --- True

8 if 125 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '5\*' + str(5) + '\*'

res = '5\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '5\*5\*' + str(5)

return '5\*5\*5'

7. remove\_digit(178, 8) = 17

1 def remove\_digit(number=178, digit=8)

2 res = 0

3 power = 1

4 while 178 > 0: --- True

5 cur\_digit = 178 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 178 // 10

number = 17

4 while 17 > 0: --- True

5 cur\_digit = 17 % 10

cur\_digit = 7

6 if 7 != 8: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 17 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 8: --- True

7 res = 7 + 1 \* 10

res = 17

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 17

Вариант: 1-1-50

1. gcd(35, 63) = 7

1 def gcd(x=35, y=63)

2 if 35 < 0: --- False

4 if 63 < 0: --- False

6 while 63 != 0: --- True

7 rem = 35 % 63

rem = 35

8 x = 63

9 y = 35

6 while 35 != 0: --- True

7 rem = 63 % 35

rem = 28

8 x = 35

9 y = 28

6 while 28 != 0: --- True

7 rem = 35 % 28

rem = 7

8 x = 28

9 y = 7

6 while 7 != 0: --- True

7 rem = 28 % 7

rem = 0

8 x = 7

9 y = 0

6 while 0 != 0: --- False

10 return 7

2. gcd(0, -32) = 32

1 def gcd(x=0, y=-32)

2 if 0 < 0: --- False

4 if -32 < 0: --- True

5 y = --32

y = 32

6 while 32 != 0: --- True

7 rem = 0 % 32

rem = 0

8 x = 32

9 y = 0

6 while 0 != 0: --- False

10 return 32

3. hex(237) = 'ED'

1 def hex(number=237)

2 if 237 == 0: --- False

4 res = ''

5 while 237 > 0: --- True

6 digit = 237 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + ''

res = 'D'

24 number = 237 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + 'D'

res = 'ED'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'ED'

4. square\_equal(-40, -66, 7) = '0.1 and -1.75'

3 def square\_equal(a=-40, b=-66, c=7)

4 if -40 != 0: --- True

5 D = -66\*-66 - 4\*-40\*7

D = 5476

6 if 5476 > 0: --- True

7 x1 = (--66 - sqrt(5476)) / (2\*-40)

x1 = 0.1

8 x2 = (--66 + sqrt(5476)) / (2\*-40)

x2 = -1.75

9 return str(0.1) + ' and ' + str(-1.75)

return '0.1 and -1.75'

5. square\_equal(-51, -72, -56) = 'no roots'

3 def square\_equal(a=-51, b=-72, c=-56)

4 if -51 != 0: --- True

5 D = -72\*-72 - 4\*-51\*-56

D = -6240

6 if -6240 > 0: --- False

10 elif -6240 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(27) = '3\*3\*3'

1 def factorize(n=27)

2 res = ''

3 while 27 > 2 and 27 % 2 == 0: --- False

6 d = 3

7 while 27 > 3: --- True

8 if 27 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 27 // 3

n = 9

7 while 9 > 3: --- True

8 if 9 % 3 == 0: --- True

9 res = '3\*' + str(3) + '\*'

res = '3\*3\*'

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

13 return '3\*3\*' + str(3)

return '3\*3\*3'

7. remove\_digit(728, 8) = 72

1 def remove\_digit(number=728, digit=8)

2 res = 0

3 power = 1

4 while 728 > 0: --- True

5 cur\_digit = 728 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 728 // 10

number = 72

4 while 72 > 0: --- True

5 cur\_digit = 72 % 10

cur\_digit = 2

6 if 2 != 8: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 72 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 8: --- True

7 res = 2 + 7 \* 10

res = 72

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 72